CPTCoder: A Reliable LLM System for Medical Procedure Code Prediction

Benlu Wang[1] **Ziyao Shangguan**[1] **Kyle Tegtmeier**[2]
Zhenyu Zhang[3] **Sophie Chheang**[2] **Arman Cohan**[1]
[1]Department of Computer Science, Yale University
[2]Department of Radiology and Biomedical Imaging, Yale School of Medicine
[3]Department of Computer Science, Stanford University
{benlu.wang, ziyao.shangguan, arman.cohan}@yale.edu
Project Homepage: https://benluwang.github.io/CPTCoder/

Abstract

We present CPTCoder, a human-in-the-loop system that predicts standardized medical procedure codes from clinical text. Clinical procedure coding is an extreme multi-label classification problem over a long-tailed space of short numeric identifiers, where a single-digit difference denotes an entirely different procedure. CPTCoder adapts an instruction-tuned LLM with a code-aware vocabulary and constrained decoding that guarantees all outputs are valid codes. To support human review, we derive per-code posterior inclusion probabilities from n-best reweighting, producing interpretable confidence scores that rank predictions and flag uncertain cases. A post-decoding constraint repair step enforces mutual-exclusion rules between conflicting codes. To enable reproducible research in this underexplored setting, we release **MIMIC-CPT** [1], a PhysioNet-accessible benchmark of 38,673 expert-cleaned report–code pairs with a deliberately hardened test split: 88% of test examples contain label combinations unseen during training, and over a third include codes with five or fewer training occurrences. We additionally provide 412,297 weakly aligned pairs and evaluate on a separate live dataset from a hospital, which includes out-of-domain radiology reports with billing-expert-verified labels. CPTCoder achieves 0.61 and 0.51 micro-F1 on the hardened MIMIC split and Hospital-298 respectively, outperforming the strongest baseline by 12 and 5 absolute points while reducing digit-level near-miss errors.

1 Introduction

Accurate procedural coding is a critical yet labor-intensive step in clinical operations and reimbursement. In many health systems, professional services are billed using Current Procedural Terminology (CPT) codes, and coders must assign multiple codes per report under detailed documentation requirements. Crucially, CPT codes are short *digit-sensitive* identifiers (e.g., `99213`), where a one-digit error can change billing meaning entirely. These properties motivate *human-in-the-loop* NLP systems that accelerate expert review rather than fully automate billing submission.

Prior work on computer-assisted coding has focused largely on ICD benchmarks (Johnson et al., 2016; Mullenbach et al., 2018), while CPT coding remains underrepresented despite its importance in professional billing workflows.

For real-world coder workflows, assistive systems must satisfy two requirements beyond raw accuracy. First, they need **controllability**: outputs should be valid CPT identifiers and robust to digit-level confusions between visually similar codes. Second, they must provide **actionable uncertainty**: per-code confidence that helps coders prioritize verification and auditing in high-stakes settings, where modern neural models can otherwise be poorly calibrated (Guo et al., 2017) and selective prediction is a natural way to defer uncertain cases to humans (Geifman and El-Yaniv, 2017).

Large instruction-tuned language models (LLMs) offer strong language understanding, but naive prompting can yield invalid formats and brittle digit-level slips. Two mechanics make this acute for CPT classification: (i) a high-cardinality, long-tailed, multi-label space where over-prediction is costly; and (ii) tokenizer granularity, where standard subword tokenizers often split a five-digit code into overlapping pieces . We therefore adapt an instruction-tuned LLM with parameter-efficient **LoRA** training (Hu et al., 2022) and introduce **Code-as-Token**, expanding the tokenizer so each CPT code becomes a single atomic token, reducing sub-token overlap and digit-level errors.

At inference time, we apply **constrained de-**

[1]We will release **MIMIC-CPT** via PhysioNet upon publication

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 605–614

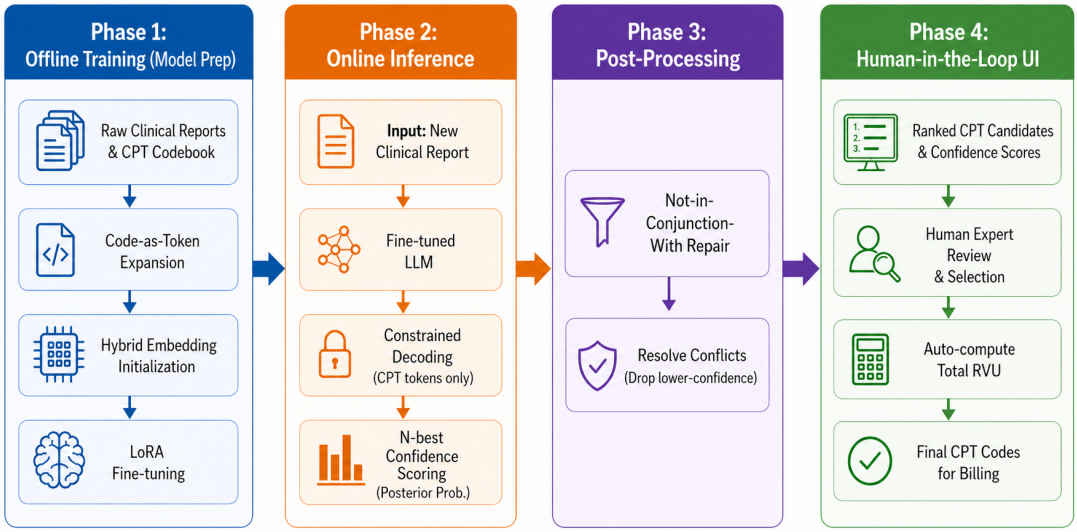

Figure 1: Overview of the CPTCoder pipeline and demo workflow.

coding to restrict generation to CPT tokens plus a small set of separators and EOS, guaranteeing syntactically valid outputs (Hokamp and Liu, 2017; Post and Vilar, 2018). To support rapid human review, we compute **code-level posterior inclusion probabilities** from an n-best constrained list—a practical approximation to posterior mass commonly used in reranking and minimum Bayes-risk style decoding (Kumar and Byrne, 2004)—and surface these probabilities as per-code confidence scores to rank candidates.

We package these components into **CPTCoder**, a web-based system for paste-and-review workflows (Figure 1). A human coder pastes a report, inspects a ranked list of candidate CPT codes with confidence, selects an appropriate subset, and obtains automatic **RVU** (Centers for Medicare & Medicaid Services) aggregation as an auxiliary signal for auditing and estimating service intensity. The model backend is deployed with **vLLM** (Kwon et al., 2023) for high concurrency and low latency (end-to-end generation typically < 10 seconds per report), making the system suitable for interactive use.

To enable reproducible research in this comparatively underexplored setting, we also introduce **MIMIC-CPT**, a PhysioNet-accessible benchmark curated from credentialed MIMIC resources: a cleaned gold split of 38,673 report–code pairs (30,938 train / 7,735 test; 350/322 unique codes) and an auxiliary silver release of 412,297 weakly aligned pairs. We additionally evaluate out-of-domain generalization on **Hospital-298**, a set of 298 de-identified radiology and interventional radiology reports whose CPT labels were verified by a billing expert. Across the hardened MIMIC test split and Hospital-298, CPTCoder achieves strong micro-F1 and markedly reduces digit-level near-miss false positives.

Contributions.

- **CPTCoder**, a parameter-efficient CPT coding system that combines **Code-as-Token** (atomic CPT tokens), **constrained decoding** for valid structured outputs, and **confidence-ranked** prediction via n-best posterior inclusion probabilities.
- **MIMIC-CPT**, a reproducible PhysioNet-accessible benchmark (clean gold split + auxiliary silver release) and **Hospital-298** for out-of-domain evaluation on radiology/IR reports.
- An **interactive web demo** that surfaces ranked CPT candidates with confidence and computes total selected-code **RVU**, deployed with **vLLM** for low-latency, high-concurrency serving.

2 Dataset Collection

While public benchmarks for medical coding predominantly target diagnosis codes (e.g., ICD), procedure coding remains underserved despite its clinical importance. Existing resources also rely on IID splits that understate real-world difficulty. We address this gap with a deliberately

challenging, reproducible benchmark. We introduce MIMIC-CPT, a **reproducible, PhysioNet-accessible** benchmark (Goldberger et al., 2000). The dataset is constructed exclusively from credentialed PhysioNet MIMIC resources (MIMIC-III (Johnson et al., 2016), MIMIC-IV (Johnson et al., 2023a,c), and MIMIC-IV-Note (Johnson et al., 2023b). We will release cohort construction scripts and CPT code-normalization utilities, and distribute processed cohorts via PhysioNet under the same access model and compliance requirements as the underlying MIMIC resources.

2.1 Gold Set

Our primary benchmark (**gold set**) is derived from MIMIC-IV-Note v2.2, which provides note/report text with comparatively direct report–code structure. We formulate each example as a multi-label instance (x_i, Y_i), where x_i is a report and Y_i is its set of CPT codes.

Cleaning. We first filter MIMIC-IV-Note to instances that contain at least one CPT code, yielding 49,341 raw report–CPT rows. Because CPT codes evolve over time, some rows contain deprecated/outdated codes. We normalize CPT codes against the most recent AMA CPT Professional Book available at the time of curation (American Medical Association, 2026) and remove rows containing deprecated/outdated CPT codes; After cleaning, the gold set contains 38,673 rows.

Training set. We construct sample-disjoint training and test partitions. The training set contains 30,938 reports and covers **350 unique CPT codes**.

Test set. We design the gold test split to be *closed-set* and deliberately more challenging than a random IID partition. We enforce $\mathcal{C}_{\text{test}} \subseteq \mathcal{C}_{\text{train}}$ and harden the test distribution toward long-tail codes and novel label co-occurrences. The resulting gold test set contains 7,735 reports and **322 unique CPT codes**.

Leakage control. Because the released silver corpus is derived from overlapping MIMIC sources, we perform strict leakage checks and remove any record-level overlaps between the gold split and the silver release (in particular, ensuring zero overlap with the gold test set). We also enforce sample-level disjointness between gold train and gold test.

Table 1: Main dataset sizes. "# codes" denotes the number of unique CPT codes in the split (gold only).

Split	# rows	# codes
Gold (IV-Note)		
Gold raw	49,341	–
Drop deprecated/old codes	10,668	–
Gold clean	**38,673**	**350**
Train	30,938	350
Test	7,735	322
Silver (III+IV)		
Silver total	412,297	–

Table 2: Silver corpus composition by label type (CPT vs. E/M).

Category	Rows
Total	412,297
CPT only	43,659 (10.59%)
E/M only	91,839 (22.27%)
CPT+E/M	276,799 (67.14%)

Silver set (auxiliary). We additionally release 412,297 weakly aligned report–code pairs from MIMIC-III/IV for future work; this silver corpus is *not used* in any experiments in this paper.

Gold hardening statistics. The full hardening breakdown for the gold split is provided in Appendix Table 4. The test set is closed-set, sample-disjoint, and deliberately shifted toward long-tail labels and novel label combinations: 38.54% of test reports contain at least one label with five or fewer training occurrences, 88.55% of test label-pair types are unseen in training, and 88.14% of test reports have an unseen full label set.

Silver label composition. The auxiliary silver release is intentionally broader and noisier than the gold set. Table 2 summarizes its CPT versus Evaluation & Management (E/M) composition.

2.2 Evaluation on External Hospital Data

We evaluate out-of-domain generalization on HOSPITAL-298, a set of 298 de-identified reports from an internal real-patient data source at a large-scale hospital. These cases primarily come from **radiology and interventional radiology (IR)** report types, and their CPT labels were annotated by a professional billing expert. We directly test the trained models on this set *without* any additional fine-tuning, threshold tuning, or k selection

on these cases, to better reflect real-world deployment performance. Results are reported together with in-domain performance in Table 3.

Expert Verification To improve label reliability, **two radiology experts** cross-checked the professional billing-expert annotations for HOSPITAL-298. A radiology expert also reviewed the CPT normalization/deprecation filtering used to curate MIMIC-CPT. These human verification steps are intended to reduce label noise in both the curated benchmark and the external evaluation set.

2.3 Usage and Evaluation

Training data. All experiments train on the gold training split; the silver corpus is *not* used for training, tuning, or evaluation.

Evaluation sets. We evaluate on (i) the in-domain *gold* test split and (ii) the out-of-domain *Hospital-298* set (§2.2), reporting direct transfer performance with no additional training or hyperparameter selection on Hospital-298.

Unified label vocabulary (365 codes). We evaluate all models over a fixed global CPT vocabulary $\mathcal{V}_{365}$ of 365 codes (from `codes_json`) to unify MIMIC and Hospital-298. MIMIC gold training covers 350 of these codes; the rest appear only in Hospital-298. Hospital-298 is used strictly for external testing.

3 Method

3.1 Overview

Given a report x, CPTCoder produces a **ranked list of candidate CPT codes** with **confidence scores** for human review. We fine-tune an instruction-tuned LLM (**Llama-3.1-8B-Instruct**) with LoRA on the gold training split using a structured target format (a sequence of CPT tokens). At inference time, we apply (i) **Code-as-Token** vocabulary expansion, (ii) **constrained decoding** to guarantee valid CPT outputs, and (iii) n-best posterior reweighting to compute **code-level confidence** used for ranking. Finally, we apply a lightweight billing-rule repair step using `not_in_conjunction_with` constraints, removing the lower-confidence code in any conflicting pair.

3.2 Code-as-Token Vocabulary Expansion

Motivation. CPT codes are short *digit-sensitive* identifiers (e.g., `99213`). With standard subword tokenizers, codes can be split into overlapping digit/substrings, which increases confusability among similar codes and can cause digit/formatting mistakes. We therefore represent each CPT code as a single *atomic* tokenizer token.

Vocabulary construction. Let $\mathcal{V}_{cpt}$ be the fixed CPT vocabulary supported by the system ($|\mathcal{V}_{cpt}| = 365$; i.e., $\mathcal{V}_{365}$ from §2.3). The MIMIC gold training split covers 350 of these codes.

For each $c \in \mathcal{V}_{cpt}$, we add one token $\tau(c) =$ `<CPT_c>` (e.g., `<CPT_99213>`) to the tokenizer so it is always treated as one token. As the CPT book updates, we can extend the vocabulary by adding new `<CPT_ · >` tokens and initializing them as below, making future codebook updates straightforward (and mitigating open-set concerns in practice).

Hybrid embedding initialization. After vocabulary expansion, we initialize each new code token embedding by blending: (i) a **code-string prior** from the raw code string and (ii) a **semantic prior** from the official code description. Let $E_{\text{in}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ be the input embedding matrix before expansion and $\mathsf{Tok}(\cdot)$ the base tokenizer. For code c with description text d_c, we compute:

$$\mathbf{v}_{\text{code}} = \frac{1}{|\mathsf{Tok}(c)|} \sum_{t \in \mathsf{Tok}(c)} E_{\text{in}}[t].$$

$$\mathbf{v}_{\text{sem}} = \frac{1}{|\mathsf{Tok}(d_c)|} \sum_{t \in \mathsf{Tok}(d_c)} E_{\text{in}}[t].$$

$$E_{\text{in}}[\tau(c)] \leftarrow \beta\, \mathbf{v}_{\text{sem}} + (1 - \beta)\, \mathbf{v}_{\text{code}}.$$

where $\beta \in [0, 1]$ controls the semantic contribution (we use β=0.85). We optionally rescale the initialized vector to match the average embedding norm of the original vocabulary to keep magnitudes well-behaved. During LoRA fine-tuning, these embeddings are updated normally.

3.3 Constrained Decoding for Structured CPT Output

To guarantee valid outputs, we restrict decoding to: (i) CPT tokens $\{\tau(c) : c \in \mathcal{V}_{cpt}\}$ in the form `<CPT_*****>`, (ii) a small set of separators (e.g., comma/newline), and (iii) EOS. This makes parsing unambiguous and prevents generating non-CPT text in the code region.

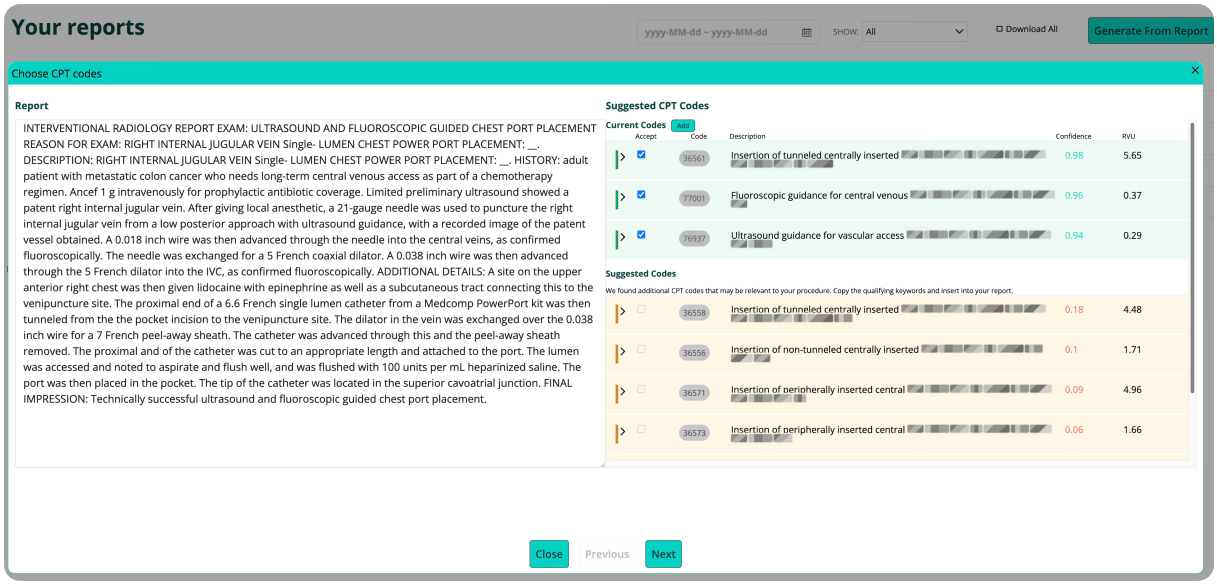

Figure 2: CPTCoder web UI. The system ranks candidate CPT codes with confidence for a given report, allowing users to confirm a final code set and view aggregated RVUs. The report shown is fully de-identified; CPT descriptions are partially masked to avoid reproducing proprietary codebook text.

3.4 n-best Decoding and Confidence Estimation

n-best generation. We run beam search to obtain B candidates $\{y_i\}_{i=1}^{B}$ and parse each candidate into a CPT sequence $\mathbf{c}_i$ (a sequence of CPT tokens).

Posterior reweighting over the n-best list. We approximate a truncated posterior over candidates by weighting each sequence using only CPT-token log-probabilities:

$$\log \tilde{w}_i = \sum_{k \in \mathcal{K}_i} \ell_{i,k}, \qquad w_i = \frac{\exp(\log \tilde{w}_i)}{\sum_{j=1}^{B} \exp(\log \tilde{w}_j)},$$

where $\mathcal{K}_i$ indexes CPT tokens in candidate i and $\ell_{i,k}$ is the token log-probability.

Code-level confidence (posterior inclusion probability). For any CPT code c, we compute its posterior inclusion probability:

$$P_{\text{set}}(c) = \sum_{i=1}^{B} w_i \cdot \mathbb{I}[c \in \text{set}(\mathbf{c}_i)] \,.$$

We return CPT candidates ranked by $P_{\text{set}}(c)$ as confidence scores in the demo UI (and optionally apply a global threshold for selection).

3.5 Not-in-Conjunction-With Constraint Repair

We extract `not_in_conjunction_with` rules from the AMA CPT Professional Book (American Medical Association, 2026) as a map $\mathcal{N}(a)$ of codes forbidden to co-occur with a. Given a predicted set $\hat{Y}$ with confidences $P_{\text{set}}(\cdot)$, we iteratively remove the lower-confidence code from any conflicting pair (a, b) where $b \in \mathcal{N}(a)$, until no conflicts remain.

4 Live System

CPTCoder is deployed as a web application that supports human-in-the-loop CPT coding from clinical reports. A user pastes a report into the website UI, and the system returns a **ranked list of candidate CPT codes** with **confidence scores** (high to low). Users can then quickly select the subset of codes they judge as correct. Figure 2 shows the deployed interface.

UI workflow. The UI exposes three core elements: (i) a report input panel, (ii) a ranked candidate list where each CPT code is accompanied by its confidence score $P_{\text{set}}(c)$, and (iii) a selection panel where users can filter/confirm codes. This design matches real coder workflows: the model proposes candidates, while the expert makes the final decision.

Automatic RVU aggregation. Given a user-selected code subset, the system computes the **total RVU** as an auxiliary auditing signal, supporting intensity checks, alternative set comparisons, and downstream billing review. We use RVU aggregation as a review aid rather than as an automatic billing decision rule.

Serving and latency. The backend uses **vLLM** on a single NVIDIA H100 (80GB) GPU for high-concurrency inference. The system is internally deployed and under evaluation by partner clinicians, with end-to-end code generation and confidence scoring typically **under 10 seconds per report**.

Human review boundary. The system is designed to accelerate expert review rather than replace it. Candidate CPT codes, confidence scores, and RVU summaries are surfaced to support coder verification, but the final code set remains a human decision.

5 Experiments

5.1 Baselines

We compare against (i) TF–IDF OvR baselines, (ii) ModernBERT multi-label classifiers, and (iii) instruction-tuned LLMs in zero-shot and LoRA fine-tuned settings.

Validation protocol. For TF–IDF and ModernBERT baselines, we use a fixed-seed split of the MIMIC training set into `train_sub/val` with `val_ratio=0.1`. Hyperparameters are selected by maximizing validation label-micro-F1. After selection, we retrain on the full training pool (`train_pool = train_sub + val`) and evaluate on (i) the MIMIC hard test split and (ii) Hospital-298 with *no additional* tuning.

Traditional multi-label classifiers (TF–IDF + OvR). We train sparse one-vs-rest (OvR) classifiers (Logistic Regression; Linear SVM) on TF–IDF features (Salton and Buckley, 1988). Predictions are decoded using either *Top-k* selection or a *global threshold* tuned on a held-out split of the MIMIC training set. For threshold decoding, we sweep $\tau \in \{0.05, 0.10, \ldots, 0.95\}$ on `val`. We select the best setting by validation label-micro-F1, refit on `train_pool`, and report results on both evaluation sets.

Encoder-based multi-label classifiers. We fine-tune `ModernBERT-{base,large}` (Warner et al., 2025) for 365-label multi-label classification and similarly select either *Top-k* or a *global threshold* on the same held-out split. For threshold decoding, we tune a global threshold $\tau \in [0.05, 0.95]$ by validation label-micro-F1. We then retrain on the full training pool and evaluate on MIMIC and Hospital-298 without additional tuning.

LLM baselines. Because MIMIC data are governed by a usage agreement that prohibits sharing with third parties, we restrict LLM experiments to strong and locally deployable open-weight models. We experiment with both *zero-shot prompting* and *parameter-efficient fine-tuning*. Zero-shot models are evaluated on MIMIC and Hospital-298 without any fine-tuning or dataset-specific tuning. For fine-tuning, we use `LLaMA-3.1-8B` with LoRA (Hu et al., 2022) and compare against our full system **CPTCoder**.

5.2 Code-as-Token Ablation

We evaluate whether *atomic* CPT tokenization (Code-as-Token) improves controllability and digit-level robustness for CPT generation (§3.2).

Ablation setting. We compare two fine-tuned LLM variants trained on the MIMIC gold training split: **BL**, LoRA fine-tuning with the *standard* tokenizer and no vocabulary expansion, and **EV**, the same model with *Code-as-Token* vocabulary expansion and hybrid initialization (§3.2). Both variants use the same constrained decoding and n-best confidence computation at inference time (§3.3–§3.4).

Near-miss FP analysis (Hamming distance). CPT codes are fixed-length 5-digit strings, so many practical errors manifest as *digit-level slips* between nearby codes. For each false-positive (FP) predicted code, we compute its minimum Hamming distance to any gold CPT code within the same report and mark it as *near-miss* if $\min d_H \leq 3$. Appendix Figure 3 shows that EV reduces the overall FP share and disproportionately reduces near-miss FPs. This supports the hypothesis that atomic CPT tokens reduce sub-token overlap, sharpen decision boundaries among visually similar codes, and improve digit-level generation fidelity.

6 Results and Discussion

Table 3 shows that **CPTCoder** achieves the best micro-F1 on both the hardened MIMIC test split and Hospital-298. On MIMIC, the strongest non-CPTCoder baseline is LoRA fine-tuned LLaMA-3.1-8B at 0.4858 micro-F1; CPTCoder reaches 0.6082, an absolute gain of 0.1224. On Hospital-298, CPTCoder improves over the same LoRA baseline from 0.4579 to 0.5094 micro-F1 despite no hospital-specific tuning. This result suggests that the code-aware generation pipeline improves

Table 3: In-domain results on the MIMIC hard test split and out-of-domain results on a real-world hospital test set. **CPTCoder** denotes our best pipeline. Model references: GPT-OSS (OpenAI, 2025), Qwen3 (Yang et al., 2025), LLaMA-3.1 (Grattafiori et al., 2024), MedGemma 1 (Sellergren et al., 2025), and MedGemma 1.5 (Sellergren et al., 2026).

Model / Setting	MIMIC (hard test)		Hospital-298	
	Micro-F1	Label-Macro-F1	Micro-F1	Label-Macro-F1
ML-based				
Sparse classifiers (TF-IDF)				
TF-IDF (OvR) + Logistic Regression, Top-k (k=3)	0.4629	0.0930	0.3919	0.0696
TF-IDF (OvR) + Linear SVM, Top-k (k=3)	0.3738	0.1103	0.1960	0.0438
TF-IDF (OvR) + Logistic Regression, Threshold	0.4350	0.0778	0.1698	0.0138
TF-IDF (OvR) + Linear SVM, Threshold	0.2216	**0.2125**	0.0987	0.0783
Encoder-based classifiers				
ModernBERT-base, Top-k (k=3)	0.4168	0.0657	0.3124	0.0470
ModernBERT-large, Top-k (k=3)	0.4782	0.1010	0.4026	0.0764
LLM (zero-shot)				
General-domain LLMs				
GPT-OSS-120B	0.0997	0.0082	0.1118	0.0226
Qwen3-30B-A3B	0.0042	0.0004	0.0062	0.0009
GPT-OSS-20B	0.0068	0.0002	0.0116	0.0020
LLaMA-3.1-8B	0.0039	0.0001	0.0161	0.0052
Medical-domain LLMs				
MedGemma-27B-Text	0.0022	0.0002	0.0050	0.0007
MedGemma-1.5-4B	0.0001	0.0000	0.0001	0.0001
LLM (fine-tuned)				
LLaMA-3.1-8B, LoRA	0.4858	0.0964	0.4579	0.2184
CPTCoder	**0.6082**	0.2046	**0.5094**	**0.2455**

not only in-domain performance but also direct transfer to out-of-domain radiology and interventional radiology reports.

The macro-F1 results highlight the remaining difficulty of long-tail CPT prediction. CPTCoder achieves the best macro-F1 on Hospital-298 and a strong macro-F1 on MIMIC, although the TF–IDF SVM threshold baseline attains slightly higher MIMIC macro-F1 by sacrificing micro-F1 substantially. This tradeoff suggests that rare-code recall remains challenging and should be evaluated separately from aggregate micro-F1. In particular, macro-F1 is sensitive to low-frequency labels, whereas micro-F1 better reflects aggregate code-instance performance.

Zero-shot open-weight LLMs perform poorly compared with supervised baselines, indicating that CPT coding cannot be solved reliably by generic prompting alone under a constrained clinical label vocabulary. Fine-tuning improves substantially, but the full CPTCoder pipeline improves further by combining code-aware tokenization, constrained structured generation, confidence-based ranking, and explicit conflict repair. These results support the design choice of treating CPT prediction as a controlled structured-generation problem rather than an unconstrained text-generation task.

Finally, the Code-as-Token ablation in Appendix Figure 3 provides a mechanistic explanation for the improvement. EV, the expanded-vocabulary Code-as-Token variant, reduces the near-miss false-positive rate per predicted code from 26.16% to 17.05%, a 9.11-point absolute reduction. In the same analysis, it also reduces the overall false-positive share from 36.75% to 24.31%. This is especially important for CPT coding, where near-neighbor codes are not harmless spelling variants but distinct billable procedures.

7 Conclusion

We present **CPTCoder**, a parameter-efficient LLM system for CPT coding that combines **Code-as-Token** with **confidence-ranked** constrained generation for human review. We also introduce **MIMIC-CPT**, a PhysioNet-accessible benchmark with a cleaned gold split, an auxiliary silver release, and a real-world radiology/IR hospital evaluation set.

Limitations

- **Coverage and domain shift.** MIMIC-CPT is derived from credentialed MIMIC resources and Hospital-298 contains $\sim$300 radiology/IR reports. Performance and error modes may differ in other specialties, outpatient settings, and coder workflows; robust cross-institution generalization remains challenging.
- **Closed-set label space (365-code vocabulary).** For unified evaluation across MIMIC and Hospital-298, CPTCoder operates on a fixed global CPT vocabulary (365 codes), while the MIMIC gold training split covers only a subset of these codes. The current system does not address truly open-set CPT prediction; codebook updates require adding new tokens and re-validating the pipeline.
- **Human-in-the-loop, not autonomous billing.** The system is designed to assist expert review by ranking candidates and exposing uncertainty. It should not be used to submit billing codes without qualified human verification. In particular, the model does not adjudicate all payer-specific policies, bundling rules, documentation requirements, or local billing practices.
- **Confidence scores are decision aids.** The posterior inclusion probabilities used by CPTCoder are derived from an n-best constrained decoding approximation. They are useful for ranking and review prioritization, but they should not be interpreted as fully calibrated probabilities without additional calibration analysis.
- **Weakly aligned auxiliary data.** The released silver corpus contains weakly aligned report–code pairs and is *not used* for training, tuning, or evaluation in this paper. Future work is needed to leverage it safely under label noise and heterogeneity.
- **External resources, licensing, and deployment constraints.** CPT descriptions and certain rule metadata used for initialization/repair may be subject to licensing constraints, which can limit what can be redistributed. In real deployments, processing clinical text requires strict institutional privacy/security controls (e.g., access control, audit logging policy, and data governance) beyond what is demonstrated in a research prototype.

Ethics Statement

We use only de-identified clinical text and follow all applicable privacy and security requirements. Use of internal hospital data was conducted under appropriate institutional oversight (with IRB approval) and institutional data governance policies. Use of MIMIC data complies with PhysioNet credentialing requirements and the MIMIC data use agreement. CPTCoder is intended for human-in-the-loop assistance rather than autonomous billing-code submission.

References

American Medical Association. 2026. *CPT 2026 Professional Edition*. American Medical Association, Chicago, IL.

Centers for Medicare & Medicaid Services. PFS Relative Value Files. https://www.cms.gov/medicare/payment/fee-schedules/physician/pfs-relative-value-files. Accessed 2026-02-27.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 4878–4887.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. 2000. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

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)*. OpenReview.net.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023a. MIMIC-IV. *PhysioNet*. Version 2.2.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023b. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet*. Version 2.2.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, and 1 others. 2023c. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-Wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180. SOSP 2023.

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.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. 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 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Andrew Sellergren, Chufan Gao, Fereshteh Mahvar, Timo Kohlberger, Fayaz Jamil, Madeleine Traverse, Alberto Tono, Bashir Sadjad, Lin Yang, and 1 others. 2026. Medgemma 1.5 technical report. *arXiv preprint arXiv:2604.05081*.

Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, and 1 others. 2025. Medgemma technical report. *Preprint*, arXiv:2507.05201.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster,

longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

An Yang and 1 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

A Additional Dataset and Error Analysis

Table 4: Gold split statistics and hardening metrics for MIMIC-CPT. The test split is sample-disjoint and closed-set ($\mathcal{C}_{\text{test}} \subseteq \mathcal{C}_{\text{train}}$), and is hardened toward long-tail codes and novel label co-occurrences. Long-tail metrics use training frequency $f_{\text{train}}(c)$.

Metric	Train	Test (hard)
# Reports	30,938	7,735
# Unique CPT codes	350	322
# Test-only CPT codes (closed-set check)	–	0
Avg. labels / report	2.17	4.05
Long-tail difficulty (training frequency f_{train})		
Share of test labels with $f_{\text{train}} > 100$	–	55.96%
Share of test labels with $f_{\text{train}} \leq 20$	–	20.30%
Test reports with any $f_{\text{train}} \leq 5$ label	–	38.54%
Test reports with any $f_{\text{train}} \leq 10$ label	–	53.38%
Test reports with any $f_{\text{train}} \leq 20$ label	–	55.31%
Compositional novelty (unseen co-occurrences / unseen label sets)		
Unseen pair types among test label pairs	–	88.55%
Multi-label test reports with any unseen pair	–	81.90%
Test reports with an unseen full label set	–	88.14%
Train/Test label JS divergence (bits)	0.3926	

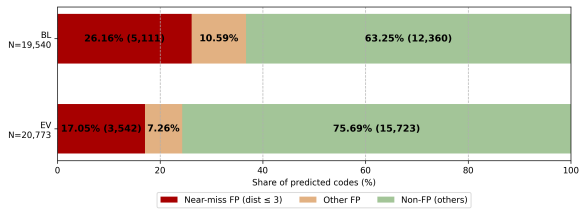

Figure 3: **Near-miss FP breakdown for BL vs. EV on the MIMIC hard test split.** BL: LoRA baseline with the standard tokenizer; EV: **Expanded Vocabulary** with hybrid initialization. N denotes the total number of predicted CPT codes. An FP is labeled *near-miss* if its minimum Hamming distance to any gold 5-digit code in the same report satisfies $\min d_H \leq 3$.

B CPT Major-Section Breakdown

Table 5: Major-section breakdown of MIMIC-CPT **gold** (train+test) by **code instances**. Counts are computed by splitting multi-code fields into individual codes and counting each code occurrence once. Total: 98,333 code instances; 350 unique codes.

Major Section	Instances	Pct	Unique codes
Surgery	45,936	46.71%	261
Radiology	36,816	37.44%	78
Anesthesia	15,137	15.39%	2
Medicine	440	0.45%	7
Evaluation & Management	3	$< 0.01\%$	1
CPT Category III	1	$< 0.01\%$	1

Table 6: Major-section breakdown of the **silver** release (`silver_raw.csv`) by **code instances**. Counts are computed by splitting `cpt_codes|em_codes` into individual codes and counting each code occurrence once. Total: 412,297 rows; 2,804,116 code instances; 3,119 unique codes.

Major Section	Instances	Pct	Unique codes
Evaluation & Management	1,587,410	56.61%	58
Surgery	694,389	24.76%	2,830
Medicine	471,404	16.81%	155
Radiology	49,481	1.76%	64
Pathology & Laboratory	1,385	0.05%	4
Other (Numeric out-of-range)	30	$< 0.01\%$	1
CPT Category III	17	$< 0.01\%$	7