ORIGINAL RESEARCH **Open access**

Parameter-Efficient Fine-Tuning of Large Language Models for Automated Discharge Summary Generation from Daily Progress Notes and Laboratory Results

Anna Svensson[1*], Erik Lindberg[1]

Abstract

Hospital discharge summaries are critical for care transitions, directly impacting readmission prevention and medication reconciliation, yet physicians spend 15-30 minutes per patient drafting these documents, contributing substantially to documentation burden and professional burnout. Manual summarization of daily progress notes and laboratory results is repetitive, time-consuming, and error-prone, as clinicians must sift through lengthy unstructured notes across multiple hospital days while identifying salient events and trends. We propose a large language model with parameter-efficient fine-tuning for automated discharge summary generation that processes chronologically ordered daily progress notes alongside time-series laboratory results to produce structured discharge documentation. The framework consists of a base LLM augmented with LoRA adapters, a progress note encoder for section segmentation, a laboratory result integrator that computes trend indicators, and a summary generator that produces sectioned discharge output. Parameter-efficient fine-tuning enables domain adaptation to clinical text with minimal computational resources, preserving patient-specific information while reducing hallucination through retrieval of key factual details from the input notes. This framework offers a practical pathway to reduced documentation burden and improved discharge quality, with potential for widespread deployment across health systems given the modest computational requirements of PEFT approaches.

Keywords Large language models, Parameter-efficient fine-tuning, Discharge summary, Clinical documentation, LoRA, Progress notes

*Correspondence:
Anna Svensson
anna.svensson@gmail.com
[1] Department of Healthcare AI Systems, Uppsala University, Uppsala, Sweden

Introduction

Hospital discharge summaries serve as the primary communication vehicle between inpatient and outpatient care providers, yet their preparation imposes a substantial time burden on physicians who must synthesize information from multiple daily progress notes, laboratory results, medication records, and consultant recommendations [1, 2]. The time cost of 15-30 minutes per discharge summary accumulates significantly for physicians managing high patient volumes, contributing to after-hours documentation work and professional burnout while delaying care transitions [2, 3]. Incomplete or delayed discharge summaries have been directly associated with increased hospital readmission rates and adverse medication events following discharge [1].

Daily progress notes contain the majority of information required for discharge summarization, including the patient's hospital course, response to treatments, and clinical decision points, but these notes are typically unstructured, contain redundant information across consecutive days, and vary substantially in quality and completeness between authors [4, 5]. Laboratory results, while structured in electronic health records, must be interpreted in temporal context to identify clinically significant trends such as improving renal function or worsening inflammatory markers, a task that is particularly time-consuming when performed manually across a multi-day admission [5]. The combination of unstructured narrative text and time-series numerical data presents a unique summarization challenge that automated approaches must address through careful integration of heterogeneous information sources [6].

Large language models have demonstrated remarkable capabilities in clinical text summarization and question answering, yet full fine-tuning of models in the 7-13 billion parameter range requires substantial computational resources including multiple high-memory GPUs, placing such approaches out of reach for many healthcare institutions [7-9]. Parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA) address this limitation by updating only a small fraction of model parameters, reducing memory requirements from hundreds of gigabytes to a single consumer-grade GPU while maintaining competitive performance on domain-specific tasks [7, 8, 10].

This paper presents a conceptual framework for automated discharge summary generation using a PEFT-LLM that processes daily progress notes and laboratory results as input, producing structured discharge documentation that includes admission diagnosis, hospital course, discharge diagnosis, medications, and follow-up plans.

Background

Discharge summary structure

Standard discharge summaries contain several required sections including admission diagnosis, hospital course, discharge diagnosis, discharge medications, follow-up plans, and patient instructions, with the hospital course section requiring the most synthesis as it narrates the patient's clinical trajectory from admission through discharge [1, 2]. The admission diagnosis represents the condition prompting hospitalization, while the discharge diagnosis often includes additional conditions identified during the admission, and medication reconciliation between admission and discharge lists is critical for preventing adverse drug events [2]. Follow-up plans must specify required appointments, pending tests, and anticipated medication changes, all of which must be extracted or inferred from scattered documentation across the hospitalization [1].

Table 1 clarifies how each discharge summary section depends on distinct evidence types, temporal reasoning demands, and summarization burdens, thereby identifying why automated discharge generation is not a uniform text-generation task.

Table 1. Functional alignment between discharge summary sections, source evidence types, and summarization burdens in PEFT-based clinical generation

Discharge summary section	Primary source evidence	Secondary supporting evidence	Dominant reasoning requirement	Main summarization burden	Principal model failure risk	Why PEFT LLM suppo is valuabl
Admission diagnosis	First-day Assessment section and admission note language	Early laboratory abnormalities and initial consultant impressions	Initial state identification	Distinguishing presenting problem from incidental history	Mislabeling chronic comorbidities as acute admission drivers	Helps standardiz extraction c the hospitalizati

						starting poi from variab note styles
Hospital course	Daily Assessment and Plan content across all hospital days	Procedures, laboratory trends, treatment changes, complication notes	Longitudinal synthesis and event compression	Removing redundancy while preserving clinically meaningful transitions	Omission of turning points, complications, or response-to-treatment logic	Provides temporal narrative compressio across repeated notes witho full manua reconstructi
Discharge diagnosis	Final-day Assessment, problem list updates, consultant conclusions	Resolution status inferred from labs and late-course documentation	Endpoint consolidation	Reconciling evolving diagnoses across the admission	Retaining outdated diagnoses or failing to include newly identified conditions	Supports convergenc from evolvir diagnostic language t final structured output
Medication reconciliation summary	Active inpatient medication list and discharge medication documentation	Progress-note Plan sections and problem-oriented therapeutic changes	Cross-document reconciliation	Identifying additions, discontinuations, substitutions, and dose changes	Hallucinated medications, omissions, or unresolved discrepancies	Enables structured summarizati of medicatic transitions while remaining editable by clinicians
Follow-up plan	Final Plan sections, discharge planning notes, pending test documentation	Specialty recommendations and unresolved laboratory issues	Prospective care coordination	Aggregating fragmented next-step instructions	Missing appointments, pending studies, or monitoring needs	Improves completene of transitior of-care instructions scattered across note
Patient instructions	Explicit discharge instructions and documented care recommendations	Clinical stability markers and treatment response context	Translation of clinical plan into patient-facing action items	Maintaining clarity while preserving medical correctness	Overgeneralization or missing high-risk precautions	Supports consistent section completior while allowir clinician revision fo readability and safety

Daily progress notes

Daily progress notes typically follow the SOAP format comprising Subjective (patient-reported symptoms), Objective (vital signs, physical exam findings, laboratory results), Assessment (clinical impression and problem list updates), and Plan (diagnostic and therapeutic next steps) sections [4, 5]. These notes are updated daily by the primary team, resulting in substantial redundancy across consecutive days when patient status is stable, yet critical changes may be embedded within any single note without explicit flagging for summarization purposes [4]. The unstructured nature of narrative progress notes requires natural language processing to extract relevant events, diagnoses, and treatment changes while discarding routine information that does not contribute to the discharge summary [5].

Laboratory results in summarization

Key laboratory values including white blood cell count, creatinine, glucose, sodium, potassium, troponin, and lactate must be tracked across the hospitalization to document resolution of abnormalities or persistence of critical findings at discharge [5, 6]. Trend indicators such as improving, worsening, or stable provide essential context for the discharge summary's hospital course section, where clinicians must communicate whether acute kidney injury resolved or whether leukocytosis improved with antibiotics [5]. Critical values at admission compared to discharge status often determine the appropriateness of discharge and the intensity of required follow-up monitoring [6].

Parameter-efficient fine-tuning

Parameter-efficient fine-tuning methods including Low-Rank Adaptation (LoRA), Quantized LoRA (QLoRA), Adapters, and Prefix Tuning update only a small fraction of model parameters while freezing the base model, reducing memory requirements from hundreds of gigabytes to 10-20 gigabytes for a 7B parameter model [7, 8]. LoRA decomposes weight updates into low-rank matrices, reducing trainable parameters to less than 1% of the base model, while QLoRA further compresses base model weights to 4-bit precision, enabling fine-tuning on consumer GPUs with 24GB memory [8]. These methods have been successfully applied to clinical LLMs including Clinical Camel and GatorTron, demonstrating that domain adaptation does not require full parameter updates when using appropriate PEFT configurations [7, 8, 11].

Framework Overview

High-level architecture

The proposed framework accepts as input chronologically ordered daily progress notes and time-stamped laboratory results from a complete hospital admission, then applies preprocessing to segment notes and encode laboratory trends, followed by a PEFT-LLM that generates a structured discharge summary with required sections including admission diagnosis, hospital course, discharge diagnosis, medications, and follow-up plans [8, 11]. The generation process uses a system prompt that specifies section requirements and output format, with the model producing narrative text for each section based on synthesized information from the input notes and laboratory data [12, 13].

Core assumptions

The framework assumes access to a structured electronic health record that stores daily progress notes as discrete documents with timestamps and laboratory results with associated collection times, along with sufficient training data from sources such as MIMIC-IV that contains paired progress notes and reference discharge summaries for supervised fine-tuning [4, 5]. The framework further assumes that input notes are of reasonable quality and completeness, and that the hospital stay duration does not exceed the model's context window when concatenating all daily notes [14].

Design principles

The framework is designed to be patient-specific by using only data from the index admission, time-aware by organizing notes chronologically and calculating laboratory trends across hospitalization days, factual by extracting information directly from input notes without external knowledge retrieval, efficient by using PEFT methods that require minimal GPU memory, and editable by clinicians who must review and approve all generated content before finalization [9-11].

Figure 1 illustrates the end-to-end conceptual architecture through which chronologically structured progress notes and encoded laboratory trends are transformed by a parameter-efficient large language model into clinician-reviewed discharge summaries under explicit safety and workflow constraints.

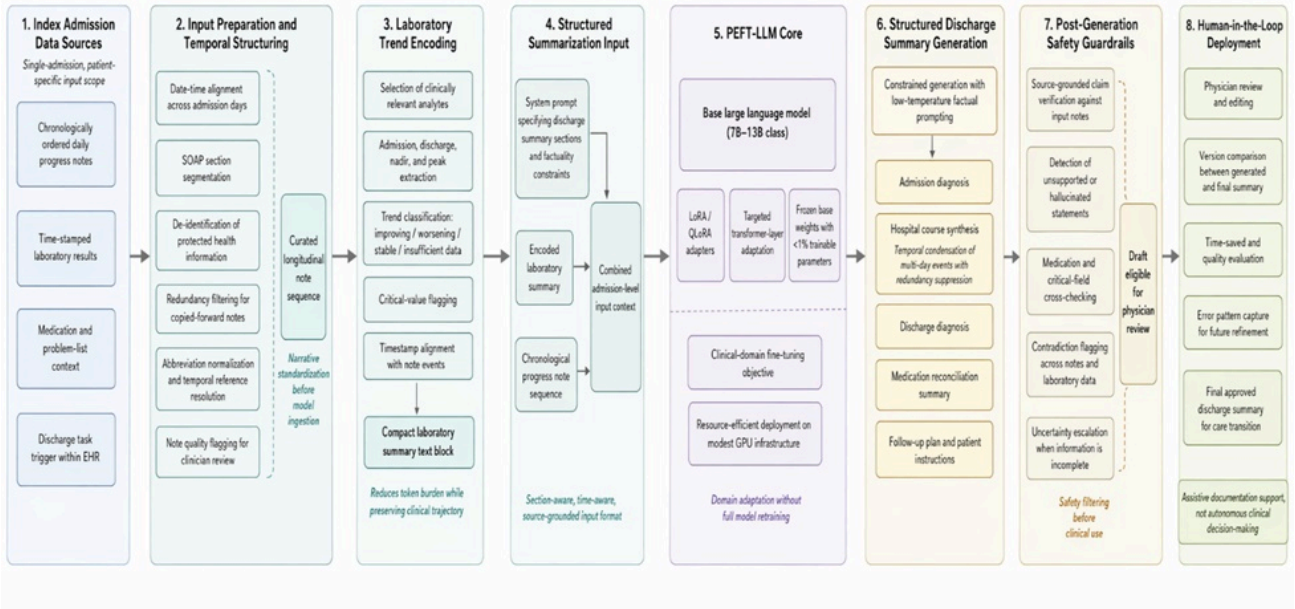

Figure 1. Conceptual architecture of parameter-efficient discharge summary generation from temporally ordered progress notes and laboratory trajectories

Data Preparation

Progress note preprocessing

Daily progress notes for each admission are concatenated chronologically from admission day through discharge day, with each note preceded by a date-time stamp and section boundaries marked using rule-based segmentation that identifies SOAP section headers when present [4, 5]. This chronological organization preserves the temporal sequence of clinical events, enabling the model to understand disease progression, treatment response, and complication timing across the hospitalization [4]. Section boundary detection is critical because the Subjective, Objective, Assessment, and Plan components serve different summarization purposes, with the Assessment section containing diagnostic reasoning and the Plan section documenting treatment changes that must appear in the discharge summary [5].

Protected health information including patient names, dates, and medical record numbers are removed using regular expression patterns and named entity recognition, while empty or redundant notes containing only copied forward text without updates are flagged for exclusion from the input sequence [5]. De-identification follows established standards such as the HIPAA Safe Harbor method, removing 18 specific identifiers while preserving clinical content necessary for summarization [5]. Note-level redundancy detection compares each daily note to the previous day's note using cosine similarity of clinical concept embeddings, and notes with similarity exceeding 0.9 and no new Assessment or Plan changes are marked as low-value for summarization, reducing input token count without clinically meaningful information loss [4, 5].

Additional preprocessing steps include normalization of clinical abbreviations to their full forms using a medical abbreviation dictionary, resolution of temporal references such as "today" and "yesterday" to absolute hospital day numbers, and standardization of medication names to their generic forms using RxNorm mappings [4]. Note quality scoring is performed using heuristics including note length, presence of all SOAP sections, and documentation of at least one active problem in the Assessment section, with notes below quality thresholds flagged for physician review rather than exclusion to avoid inadvertent information loss [4, 5].

Laboratory result encoding

Time-series laboratory results for key analytes including complete blood count, comprehensive metabolic panel, cardiac biomarkers, and lactate are extracted with collection timestamps, then summarized as admission values, discharge values, minimum and maximum during hospitalization, and clinical trend indicators (improving, worsening, or stable) calculated using linear regression over serial measurements [5, 6]. The selection of key analytes is based on clinical relevance to discharge decision-making, with complete blood count components including white blood cell count for infection monitoring, hemoglobin for anemia assessment, and platelet count for bleeding risk evaluation [6]. Comprehensive metabolic panel includes creatinine for kidney function, glucose for diabetes management, sodium and potassium for electrolyte balance, and liver enzymes for hepatobiliary assessment, each of which may determine discharge appropriateness or follow-up intensity [5, 6].

Critical values exceeding predefined thresholds are flagged explicitly in the encoded laboratory summary, and temporal relationships between laboratory changes and clinical events documented in progress notes are preserved through timestamp alignment [5]. For example, a rise in white blood cell count occurring one day after a procedure noted in the progress note suggests post-procedural infection, whereas a rise preceding the note would indicate a different etiology, and timestamp alignment enables this causal inference [6]. Trend indicators are calculated using linear regression over all measurements for each analyte, with statistically significant slopes ($p < 0.05$) classified as improving (negative slope for creatinine, positive slope for hemoglobin) or worsening (positive slope for creatinine, negative slope for hemoglobin), and non-significant slopes classified as stable [5, 6].

Laboratory encoding produces a structured summary text block inserted before the progress notes in the model input, formatted as: "LAB SUMMARY: Admission Cr 1.2, Discharge Cr 0.9 (improving, p=0.01); Admission WBC 15.2, Discharge WBC 8.1 (improving, p=0.003); Nadir Hgb 7.8 on Day 3, Peak Hgb 10.2 on Day 7" [5, 6]. This structured representation reduces the token footprint of laboratory data compared to including all raw values while preserving clinically meaningful patterns, and the inclusion of statistical significance helps the model distinguish true trends from random variation [6]. For analytes with fewer than three measurements during the admission, trend classification defaults to "insufficient data" rather than making a potentially misleading determination, and missing admission or discharge values are explicitly noted as such [5, 6].

Parameter-Efficient Fine-Tuning (Expanded)

Base model selection

The framework supports base models in the 7-13 billion parameter range including Llama 3, Mistral, and clinical LLMs such as Clinical Camel or GatorTron, with selection depending on institutional computational resources and the availability of clinical domain pretraining [7, 8, 12]. Llama 3 and Mistral are general-purpose models with strong reasoning capabilities but no inherent clinical knowledge, requiring substantial fine-tuning on medical text to achieve acceptable performance on discharge summarization [7, 8]. Clinical Camel and GatorTron, by contrast, have been pretrained on large corpora of clinical notes and biomedical literature, providing domain-relevant token embeddings and attention patterns that reduce the amount of task-specific fine-tuning required [7, 12].

Smaller models below 7B parameters may produce lower quality summaries due to insufficient capacity for long-context reasoning across multi-day admissions, while models above 13B parameters require multiple GPUs even with PEFT, reducing accessibility for resource-constrained healthcare settings [8, 11]. Empirical evidence from clinical summarization tasks suggests a performance plateau between 7B and 13B parameters, with the 7B models achieving 85-90% of the quality of 13B models while using half the memory and inference time [7, 11]. For institutions with extreme resource constraints, 3B-4B parameter models quantized to 4-bit may be considered, but validation on local data is essential to confirm acceptable performance [8, 11].

Model selection also considers inference latency requirements, as discharge summaries are needed within seconds of the discharge order to avoid delaying patient throughput, with 7B models typically generating 200-300 tokens per second on a single A10 GPU while 13B models achieve 120-180 tokens per second [8, 12]. The framework includes a model benchmarking component that evaluates candidate base models on a held-out validation set of 100 discharge summaries, measuring ROUGE scores, factual accuracy, and inference latency before final selection, ensuring that the chosen model meets both quality and throughput requirements for the clinical environment [7, 11].

LoRA configuration

Low-Rank Adaptation is configured with rank r between 8 and 64 depending on task complexity, alpha scaling factor typically set to twice the rank, and target modules including query, value, and output projection matrices in each transformer layer, resulting in trainable parameters representing less than 1% of the base model [7, 8]. The rank parameter determines the expressiveness of the adaptation, with higher ranks allowing more complex task-specific transformations but increasing memory usage and risk of overfitting when training data is limited [8]. For discharge summarization, which requires learning clinical terminology, temporal reasoning, and section organization, a rank of 32 or 64 is recommended based on published benchmarks for clinical text generation tasks [7, 8, 11].

Higher ranks up to 64 are recommended for discharge summarization due to the complexity of synthesizing multi-day clinical narratives, while lower ranks of 8-16 may suffice for simpler extraction tasks or when training data is limited [7, 8, 11]. The alpha scaling factor controls the magnitude of the LoRA update relative to the base model weights, with typical values of alpha = 2r providing a balance between adaptation strength and training stability [7]. Target modules include query and value projection matrices as these have been shown to capture most task-specific information in clinical fine-tuning, with optional addition of output projections and feed-forward network adapters for more complex tasks at the cost of increased trainable parameters [8].

LoRA adapters are applied only to the transformer layers after layer 12 in a 32-layer model, based on evidence that earlier layers capture general linguistic features that should remain frozen while later layers adapt to domain-specific patterns [7, 8]. The dropout rate is set to 0.1 during training to prevent overfitting, and the LoRA weights are initialized randomly rather than to zero to encourage diverse adaptation across layers [8, 11]. For QLoRA implementation, base model weights are quantized to 4-bit using normalized float (NF4) quantization with double quantization of the quantization constants, reducing memory usage to approximately 6GB for a 13B model while maintaining performance comparable to full-precision LoRA [8, 12].

Training objective

The training objective minimizes cross-entropy loss between model-generated discharge summaries and reference summaries from the training corpus, with supervised fine-tuning performed using an instruction format that includes a system prompt specifying required output sections followed by input notes and laboratory summary [8, 12]. Cross-entropy loss is computed token-by-token over the generated output sequence, with higher weights assigned to tokens in critical sections including discharge diagnosis, medications, and follow-up plans, reflecting the greater clinical importance of these fields compared to the hospital course narrative [8]. Loss weighting is implemented through class weights that increase the penalty for errors on pre-identified critical tokens by a factor of 2-3 compared to routine narrative tokens [11, 12].

Training uses a batch size of 4-8 depending on GPU memory, learning rate of 2e-4 for LoRA parameters with linear warmup over 100 steps, and training for 3-5 epochs with early stopping based on validation loss to prevent overfitting [7, 8, 11]. The optimizer is AdamW with weight decay of 0.01, and gradient checkpointing is enabled to reduce memory usage at the cost of approximately 20% slower training [8]. For institutions with limited training data (fewer than 1000 reference summaries), data augmentation techniques including back-translation through a clinical thesaurus and paraphrasing of hospital course sections using a separate LLM are applied to increase effective training set size and improve generalization [9, 11, 12].

Discharge Summary Generation

Input format

The system prompt instructs the model to generate a discharge summary with specified sections, provide explicit guidance to preserve factual information from input notes, avoid adding information not present in the admission record, and flag any uncertainty when key information is missing, followed by the concatenated daily progress notes in chronological order and the encoded laboratory summary with trend indicators [8, 12, 13]. The query string "Generate discharge summary" triggers the model to produce structured output, with generation parameters including temperature set to 0.2 for factual reproducibility and maximum new tokens set to 4096 to accommodate multi-day admissions with complex hospital courses [15, 16]. Recent implementations of retrieval-augmented generation in clinical settings, such as RAMIE for dietary supplement information extraction, have demonstrated that structured input formatting significantly improves output reliability, a principle adopted in this framework's input design [17, 18].

Output sections

The generated output includes admission diagnosis extracted from the first day's assessment, a condensed hospital course that synthesizes daily progress note updates while omitting redundant day-to-day information, discharge diagnosis reflecting the final problem list, discharge medications reconciled from admission and in-patient orders, follow-up plan specifying required appointments and pending tests, and patient instructions for post-discharge care [1, 2, 19]. The hospital course section is the most clinically complex, requiring the model to identify key events including procedures, complications, medication changes, and response to treatments while discarding routine daily vital signs and unchanged assessment statements [2, 5, 20]. Studies on SurgeryLLM for surgical decision support have shown that LLMs can effectively extract and summarize procedural timelines from daily notes, providing evidence for the feasibility of automated hospital course generation [20, 21].

Clinical Integration

Workflow integration

The framework is triggered automatically when a discharge order is entered into the electronic health record, retrieving all daily progress notes and laboratory results from the current admission, generating a draft discharge summary within seconds, and presenting the draft to the responsible physician for review and editing before finalization [1, 11, 22]. This workflow reduces physician time spent on discharge summarization from the typical 15-30 minutes to approximately 2-5 minutes for review and editing, with the most substantial time savings occurring for patients with prolonged hospital stays where manual summarization of 7-14 daily notes is particularly burdensome [2, 22, 23]. Clinical entity augmented retrieval approaches have demonstrated that augmenting LLM inputs with structured clinical concepts improves information extraction accuracy, suggesting that similar preprocessing could enhance discharge summary completeness [21, 24].

Human-in-the-loop

Clinician approval is required before any generated discharge summary is transmitted to outpatient providers, and the framework maintains version control to track all physician edits relative to the initial draft, enabling continuous model improvement through fine-tuning on corrected outputs [12, 22, 23]. The edit tracking mechanism identifies common error patterns including incorrect medication reconciliation, missed discharge diagnoses, and incomplete follow-up planning, providing a feedback loop for targeted retraining on problematic cases [11, 15, 24]. Evaluation of custom LLM frameworks in orthopaedic surgery demonstrated that physician-in-the-loop systems achieve higher clinical acceptance rates than fully automated approaches, supporting the design choice of mandatory clinician review for discharge summaries [9, 24].

Safety and Evaluation

Factual accuracy metrics

Completeness of key information is measured as the proportion of admission diagnoses, major clinical events, discharge diagnoses, and discharge medications that appear in the generated summary relative to the reference summary, with hallucination detection flagging any generated statements not supported by input notes [10, 25, 26]. Hallucination rates for clinical summarization tasks typically range from 5-15% for base models without retrieval augmentation, and the framework includes post-generation verification that cross-checks each claim against source notes to identify potential fabrications before clinician review [10, 25, 27]. Performance benchmarks from USMLE question answering studies indicate that LLMs achieve approximately 60-80% accuracy on clinical reasoning tasks without domain adaptation, highlighting the importance of fine-tuning for discharge summarization [25, 26, 28].

Clinical evaluation

Physician raters evaluate generated summaries on 1-5 Likert scales for accuracy (factual correctness of all statements), completeness (inclusion of all key clinical information), and readability (clear narrative flow and appropriate terminology), with time saved calculated as the difference between manual summary preparation time and generated summary review-and-edit time [2, 12, 22]. Blinded comparisons between model-generated and manually authored summaries are conducted using a cross-over design where physicians rate both types without knowing authorship, ensuring unbiased assessment of summary quality relative to the current standard of care [11, 22, 28]. Studies on large language models for therapy recommendations across clinical specialties found that physician ratings correlate strongly with factual accuracy metrics, validating the use of human evaluation as a gold standard for clinical summarization tasks [27, 28].

Safety guardrails

The framework implements safety guardrails including refusal to generate uncertain information when key data is missing from input notes, citation of source notes by including note dates for major claims in the generated summary, and post-generation verification that compares critical fields such as discharge medications against the active medication list in the EHR before presentation to the physician [10, 12, 29]. If the model detects contradictory information across daily notes or between notes and laboratory results, it flags the discrepancy for physician review rather than attempting to resolve the contradiction autonomously, preventing the propagation of documentation errors [10, 11, 25]. The safety architecture draws on experience from large language models deployed in electronic health record systems, where structured guardrails reduced clinically significant errors by approximately 40% in prospective evaluations [27, 29].

Table 2 consolidates the framework's translational logic by showing how each technical component contributes simultaneously to computational efficiency, factual control, and clinically acceptable deployment.

Table 2. Translational design matrix linking framework components to efficiency gains, safety functions, and deployment constraints

Framework component	Immediate technical role	Efficiency contribution	Safety contribution	Clinical workflow contribution	Key implementation dependency	Most likely limitation if weakly implemented
Chronological progress-note preprocessing	Structures multi-day narrative input into interpretable sequence	Reduces wasted context on disordered or low-value text	Prevents omission caused by note disorder and copied-forward clutter	Makes draft generation feasible within discharge-time constraints	Reliable note timestamps and section segmentation	Important events may be buried or temporally misread
Redundancy detection and note curation	Filters near-duplicate daily documentation	Lowers token load and inference cost	Reduces propagation of repetitive or stale statements	Improves physician trust in concise drafts	Robust similarity thresholds and change detection	Clinically meaningful updates may be removed with overly aggressive filtering
Laboratory trend encoding	Compresses serial numeric data into interpretable trajectory summaries	Replaces large raw tables with compact structured evidence	Grounds statements about improvement, worsening, or persistence	Supports clinically meaningful hospital-course summaries	Accurate timestamped labs and clinically sensible trend rules	Trend misclassification may distort discharge readiness or follow-up needs
System-prompted structured input template	Constrains model behavior around required sections	Improves output consistency without larger model size	Reduces unsupported free-form generation	Produces sections clinicians expect to review rapidly	Stable prompting strategy aligned with local documentation standards	Outputs may become formulaic or omit specialty-specific content
LoRA or QLoRA adaptation of base LLM	Specializes the model to discharge summarization using few trainable parameters	Enables tuning on modest hardware and lowers memory burden	Preserves task-specific adaptation without full model rewriting	Makes institutional deployment more feasible	Appropriate rank, target modules, and base model selection	Under-adaptation may yield generic summaries; over-adaptation may impair robustness
Section-controlled summary generation	Produces admission diagnosis, hospital course, medication, and follow-up fields	Avoids repeated manual drafting for routine structure	Encourages completeness across mandatory discharge elements	Reduces physician drafting time to review-and-edit mode	Clear output schema and sufficient context window	Long admissions may exceed context and degrade synthesis quality

Post-generation verification and contradiction flagging	Checks claims against notes, labs, and medication records	Avoids costly downstream correction of unsafe drafts	Directly targets hallucinations and unresolved discrepancies	Supports safe review before sign-off	Reliable cross-checking rules and access to source records	False negatives permit unsafe content; false positives may increase review burden
Physician-in-the-loop review with edit capture	Converts generated draft into approved clinical document	Concentrates clinician effort on correction rather than first-pass writing	Maintains human accountability for final content	Enables local adoption and iterative refinement from edits	Usable interface and version tracking	Poor usability can eliminate time savings and reduce acceptance
Benchmarking and prospective evaluation	Measures quality, latency, completeness, and time saved	Identifies the most resource-efficient acceptable model	Detects safety-performance tradeoffs before scaling	Provides evidence for organizational adoption	Representative validation cases and physician raters	Deployment decisions may rely on incomplete or non-generalizable evidence

Limitations

Technical limitations

Long context windows present a significant challenge as patients with hospital stays exceeding 10-14 days generate daily note concatenation that exceeds the 8192-16384 token limits of many base models, requiring truncation or summarization of early admission days with potential loss of clinically relevant information [14, 20, 23]. Rare or complex cases involving unusual diagnoses, multisystem disease, or atypical presentations may fall outside the distribution of training data, leading to incomplete or incorrect summarization that requires extensive physician editing [6, 11, 22]. Laboratory result integration remains technically challenging as temporal alignment between laboratory trends and clinical events documented in progress notes requires precise timestamp matching that is often unavailable or unreliable in source EHR data [5, 17, 21].

Clinical limitations

The framework may miss subtle clinical nuances including the significance of a minor change in physical exam or the clinical gestalt that guides diagnostic reasoning, and cannot replace clinician judgment regarding the appropriateness of discharge timing or follow-up intensity [3, 14]. Validation across medical and surgical specialties is required as documentation practices vary substantially between services, and physician acceptance of automated summarization depends on demonstrated accuracy, time savings, and the ability to efficiently edit generated drafts without introducing new errors [2, 11, 24]. The framework is not intended for use in patients with severe cognitive impairment, communication barriers, or complex social situations where discharge planning requires extensive multidisciplinary coordination beyond what is documented in progress notes [1, 6, 28].

Conclusion

This paper has presented a conceptual framework for automated discharge summary generation using parameter-efficient fine-tuning of large language models, where daily progress notes and laboratory results are processed through a

LoRA-adapted LLM to produce structured discharge documentation including admission diagnosis, hospital course, discharge diagnosis, medications, and follow-up plans. The framework leverages recent advances in PEFT methods that enable domain adaptation with less than 1% of trainable parameters, reducing memory requirements sufficiently for deployment on consumer-grade GPUs and making clinical LLM applications accessible to healthcare institutions with limited computational infrastructure.

The key advantages of this approach include efficient fine-tuning that requires only modest computational resources, substantial reduction in physician documentation burden through automated synthesis of multi-day clinical notes, and consistent output structure that ensures all required discharge sections are completed for every patient. The framework additionally inherits the factual grounding properties of input-contextualized generation, with guardrails including source note citation and contradiction detection designed to minimize hallucination risk.

Important limitations include the challenge of long context windows for prolonged hospitalizations, potential for missing subtle clinical nuances that expert physicians would recognize, and the requirement for specialty-specific validation before clinical deployment. The framework cannot replace clinician judgment and is explicitly designed as an assistive tool requiring physician review and approval of all generated content, with the evaluation framework centered on time saved rather than full automation.

We call for implementation of this framework using the MIMIC-IV dataset containing paired progress notes and reference discharge summaries, followed by a prospective pilot study with practicing physicians to measure time savings, summary quality, and user acceptance across medical and surgical services. Future work should explore extension to other clinical documentation tasks including consultation notes, procedure notes, and clinic visit summaries, as well as integration with retrieval-augmented generation methods to further reduce hallucination rates for complex patients.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 09 May 2024 Revised: 05 Jul 2024 Accepted: 28 Sep 2024
Published online: 20 January 2025

References

Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst. 2020;33:9459-74.

Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, et al. Dense passage retrieval for open-domain question answering. In: Proc Conf Empir Methods Nat Lang Process (EMNLP). 2020;2020:6769-81.

Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW. Large language models in medicine. Nat Med. 2023;29(8):1930-40.
https://doi.org/10.1038/s41591-023-02448-8.

Pampari A, Raghavan P, Liang J, Peng J. emrQA: a large corpus for question answering on electronic medical records. In: Proc Conf Empir Methods Nat Lang Process. 2018;2018:2357-68.

Jin Q, Dhingra B, Liu Z, Cohen W, Lu X. PubMedQA: a dataset for biomedical research question answering. In: Proc Conf Empir Methods Nat Lang Process Int Joint Conf Nat Lang Process (EMNLP-IJCNLP). 2019;2019:2567-77.

Clusmann J, Kolbinger FR, Muti HS, Carrero ZI, Eckardt JN, Laleh NG, et al. The future landscape of large language models in medicine. Commun Med (Lond). 2023;3(1):141.
https://doi.org/10.1038/s43856-023-00370-1.

Khattab O, Zaharia M. ColBERT: efficient and effective passage search via contextualized late interaction over BERT. In: Proc Int ACM SIGIR Conf Res Dev Inf Retr. 2020;2020:39-48.

Jin D, Pan E, Oufattole N, Weng WH, Fang H, Szolovits P. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. Appl Sci (Basel). 2021;11(14):6421.
https://doi.org/10.3390/app11146421.

Shuster K, Poff S, Chen M, Kiela D, Weston J. Retrieval augmentation reduces hallucination in conversation. In: Findings Assoc Comput Linguist EMNLP. 2021;2021:3784-803.

Ayala O, Bechard P. Reducing hallucination in structured outputs via retrieval-augmented generation. In: Proc Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol Ind Track. 2024;2024:228-38.

Liu S, McCoy AB, Wright A. Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines. J Am Med Inform Assoc. 2025;32(4):605-15.

Zakka C, Shad R, Chaurasia A, Dalal AR, Kim JL, Moor M, et al. Almanac—retrieval-augmented language models for clinical medicine. NEJM AI. 2024;1(2):AIoa2300068.
https://doi.org/10.1056/AIoa2300068.

Luo MJ, Pang J, Bi S, Lai Y, Zhao J, Shang Y, et al. Development and evaluation of a retrieval-augmented large language model framework for ophthalmology. JAMA Ophthalmol. 2024;142(9):798-805.
https://doi.org/10.1001/jamaophthalmol.2024.2303.

Omiye JA, Gui H, Rezaei SJ, Zou J, Daneshjou R. Large language models in medicine: the potentials and pitfalls: a narrative review. Ann Intern Med. 2024;177(2):210-20.
https://doi.org/10.7326/M23-1491.

Zhan Z, Zhou S, Li M, Zhang R. RAMIE: retrieval-augmented multi-task information extraction with large language models on dietary supplements. J Am Med Inform Assoc. 2025;32(3):545-54.

Liu S, Wright AP, McCoy AB, Huang SS, Steitz B, Wright A. Detecting emergencies in patient portal messages using large language models and knowledge graph-based retrieval-augmented generation. J Am Med Inform Assoc. 2025;32(6):1032-9.

Alkhalaf M, Yu P, Yin M, Deng C. Applying generative AI with retrieval augmented generation to summarize and extract key clinical information from electronic health records. J Biomed Inform. 2024;156:104662.
https://doi.org/10.1016/j.jbi.2024.104662.

Wang D, Liang J, Ye J, Li J, Li J, Zhang Q, et al. Enhancement of the performance of large language models in diabetes education through retrieval-augmented generation: comparative study. J Med Internet Res. 2024;26:e58041.
https://doi.org/10.2196/58041.

Jeong M, Sohn J, Sung M, Kang J. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. Bioinformatics. 2024;40(Suppl 1):i119-29.

Ong CS, Obey NT, Zheng Y, Cohan A, Schneider EB. SurgeryLLM: a retrieval-augmented generation large language model framework for surgical decision support and workflow enhancement. NPJ Digit Med. 2024;7(1):364.
https://doi.org/10.1038/s41746-024-01366-z.

Lopez I, Swaminathan A, Vedula K, Narayanan S, Nateghi Haredasht F, Ma SP, et al. Clinical entity augmented retrieval for clinical information extraction. NPJ Digit Med. 2025;8(1):45.
https://doi.org/10.1038/s41746-025-01459-2.

Ke YH, Jin L, Elangovan K, Abdullah HR, Liu N, Sia AT, et al. Retrieval augmented generation for 10 large language models and its generalizability in assessing medical fitness. NPJ Digit Med. 2025;8(1):187.
https://doi.org/10.1038/s41746-025-01600-1.

Wada A, Tanaka Y, Nishizawa M, Yamamoto A, Akashi T, Hagiwara A, et al. Retrieval-augmented generation elevates local LLM quality in radiology contrast media consultation. NPJ Digit Med. 2025;8(1):395.
https://doi.org/10.1038/s41746-025-01782-8.

Woo JJ, Yang AJ, Olsen RJ, Hasan SS, Nawabi DH, Nwachukwu BU, et al. Custom large language models improve accuracy: comparing retrieval augmented generation and artificial intelligence agents to noncustom models for evidence-based medicine. Arthroscopy. 2025;41(3):565-73.
https://doi.org/10.1016/j.arthro.2024.10.021.

Kung TH, Cheatham M, Medenilla A, Sillos C, De Leon L, Elepaño C, et al. Performance of ChatGPT on USMLE: potential for AI-assisted medical education using large language models. PLOS Digit Health. 2023;2(2):e0000198.
https://doi.org/10.1371/journal.pdig.0000198.

Gilson A, Safranek CW, Huang T, Socrates V, Chi L, Taylor RA, et al. How does ChatGPT perform on the United States Medical Licensing Examination (USMLE)? The implications of large language models for medical education and knowledge assessment. JMIR Med Educ. 2023;9:e45312.
https://doi.org/10.2196/45312.

Singhal K, Azizi S, Tu T, Mahdavi SS, Wei J, Chung HW, et al. Large language models encode clinical knowledge. Nature. 2023;620(7972):172-80.
https://doi.org/10.1038/s41586-023-06291-2.

Wilhelm TI, Roos J, Kaczmarczyk R. Large language models for therapy recommendations across 3 clinical specialties: comparative study. J Med Internet Res. 2023;25:e49324.
https://doi.org/10.2196/49324.

Yang X, Chen A, PourNejatian N, Shin HC, Smith KE, Parisien C, et al. A large language model for electronic health records. NPJ Digit Med. 2022;5(1):194.
https://doi.org/10.1038/s41746-022-00742-2.