SplitThenMerge: Token-Level Skill-Compositional Sparse Mixture-of-Experts for Complex Domain-Specific Tasks

Yuting Huang[1,2], **Jiawen Zhang**[1], **Yiquan Wu**[1*], **Yinghao Hu**[1], **Fei Wu**[1], **Kun Kuang**[1*]

[1]Zhejiang University, Hangzhou, China
[2]Fashton Technology, Hangzhou, China
{yutinghuang, zhangjiawenzju, wuyiquan, huyinghao, wufei, kunkuang}@zju.edu.cn

Abstract

Large language models have demonstrated strong performance on general-purpose tasks but often fail to satisfy the accuracy requirements of knowledge-intensive domains such as law, medicine, and finance. Complex domain-specific generation is inherently compositional, involving multiple atomic skills such as reasoning, knowledge grounding, and numerical computation that are frequently interleaved at the token level. Existing domain adaptation methods typically train these heterogeneous skills jointly within a single objective, which makes it difficult for models to reliably coordinate multiple skills when solving complex tasks. In this work, we explicitly incorporate atomic skills into domain-specific model training and propose SplitThenMerge, a framework that decomposes domain competence into atomic skills, trains them independently, and composes them dynamically during generation. SplitThenMerge adopts a token-level sparse Mixture-of-Experts architecture to enable fine-grained skill routing and coordination while implementing each skill as a lightweight LoRA expert to achieve parameter-efficient specialization. Experimental results demonstrate that our method consistently achieves superior performance in both legal and medical domains under the same training parameter budget.

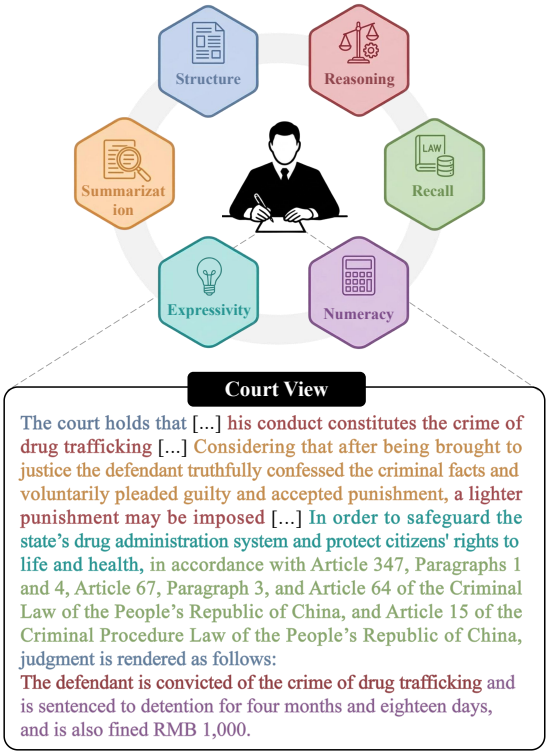

Figure 1: Drafting a court view requires legal professionals to frequently switch among structure, reasoning, summarization, expressivity, recall, and numeracy at a fine-grained, token-level resolution.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across a wide range of general-purpose language tasks, including open-ended question answering, dialogue, and reasoning benchmarks (Guo et al., 2025; OpenAI, 2025). However, their direct application to knowledge-intensive domains such as law, medicine, and finance often fails to meet practical requirements. Prior studies show that general-purpose models frequently suffer from hallucinations, unstable reasoning, and omission of critical details in professional settings (Dahl et al., 2024; Handler et al., 2025). Consequently, constructing high-performance domain-specific LLMs remains a fundamental challenge.

A key reason for this limitation is that complex domain tasks are inherently compositional rather than monolithic. In domain-specific settings, a single output often requires the coordinated use of multiple distinct abilities, yet existing domain adaptation methods typically train a single model to absorb all capabilities implicitly. Our observations indicate that domain-specific generation can be systematically decomposed into a set of atomic skills. As illustrated in Figure 1, legal judgment

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 12471–12483

generation interleaves structure, reasoning, knowledge, numeracy, expressivity, and summarization, where different words within the same sentence can correspond to different skills. Domain professionals naturally alternate among these skills during writing, suggesting that effective domain modeling requires fine-grained, token-level specialization. This observation directly motivates a token-level sparse Mixture-of-Experts (MoE) formulation, in which different skills can be selectively activated during generation.

Despite these insights, realizing a skill-oriented, token-level Mixture-of-Experts (MoE) paradigm for domain adaptation remains challenging. **From the training perspective, experts must possess clearly separated capabilities while still collaborating within a single model.** This requires constructing skill-aligned supervision to enable independent expert training and integrating these experts without compromising the integrity of the underlying model. **From the architectural perspective, domain-specific generation demands frequent and precise skill switching**, which calls for token-level routing rather than coarse-grained expert selection.

To address these challenges, we propose SplitThenMerge, a framework that explicitly operationalizes domain compositionality through token-level atomic skills. SplitThenMerge consists of three stages. First, in the expert decomposition stage, we automatically construct skill-specific training data by decomposing domain-specific fine-tuning corpora into token-level atomic skill segments. Second, in the expert training stage, we train a set of lightweight LoRA-based atomic experts, each specializing in a single skill, together with a global router that learns to perform token-level skill selection. Finally, in the expert merging stage, the independently trained experts are integrated into a unified token-level sparse MoE model, enabling implicit collaboration among experts through shared representations during inference.

This design offers advantages in both performance and efficiency. By explicitly specializing atomic skills, SplitThenMerge reduces cross-skill interference commonly observed in dense fine-tuning, while fine-grained, token-level skill coordination further enables the model to invoke appropriate capabilities at each generation step. At the same time, independent PEFT-based expert training reduces the number of trainable parameters. We empirically validate these advantages through experiments on legal judgment generation and medical discharge note generation. Under the same training parameter budget, SplitThenMerge consistently achieves stronger performance.

The contributions of this paper can be summarized as follows:

1. We analyze complex domain-specific generation and show that such tasks can be decomposed into a set of fine-grained, token-level interleaved atomic skills, motivating a compositional view of domain competence.
2. Based on this perspective, we introduce a novel training paradigm and atomic-expert architecture. Our method SplitThenMerge automatically constructs skill-specific training data from SFT corpora, independently trains experts and a global router, and merges them into a unified sparse MoE.
3. Extensive experiments demonstrate that decomposing domain-specific generation into six atomic skills leads to consistent performance improvements in legal and medical domains, while the proposed method itself is not limited to this particular skill set.

2 Related Work

Recent work has mainly focused on domain-specific LLMs and sparse MoE architectures.

2.1 Domain-Specific Large Language Models

In knowledge-intensive domains, a growing body of work has demonstrated the effectiveness of supervised fine-tuning on domain-specific data across a diverse set of tasks. In the legal domain, models such as DISC-LawLLM (Yue et al., 2024) and ChatLaw (Cui et al., 2024) construct task-oriented datasets covering judgment prediction, judicial summarization, legal examinations, and legal question answering, and achieve more reliable performance through domain-specific SFT. Similarly, in the medical domain, BioGPT (Luo et al., 2022) is pre-trained exclusively on biomedical literature, while ChatDoctor (Li et al., 2023) is further fine-tuned on doctor–patient dialogue transcripts to better handle clinical questioning and conversational diagnosis, highlighting the importance of task-aligned supervision. In the financial domain, FinGPT (Liu et al., 2023) collects internet-scale public financial texts and builds datasets for tasks such as financial headline analysis, financial infor-

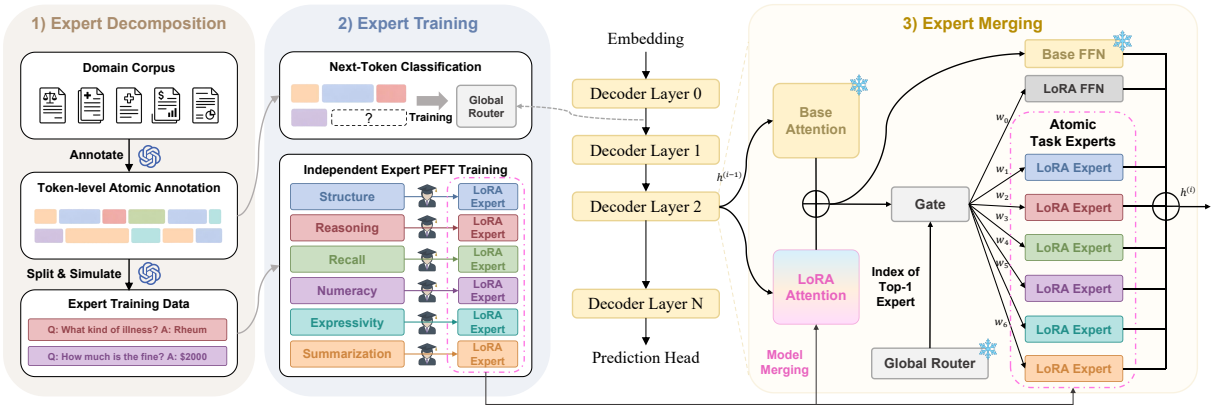

Figure 2: The proposed SplitThenMerge framework. It consists of (1) expert decomposition via token-level atomic annotation and expert-specific data construction, (2) expert training of a global router and atomic LoRA experts, and (3) expert merging into atomic-expert decoder layers with token-level expert activation selected by the global router.

mation extraction, and financial question answering, leveraging parameter-efficient techniques such as LoRA (Hu et al., 2022) to inject domain knowledge effectively.

Collectively, these domain-specific LLMs demonstrate that with appropriate data construction and fine-tuning strategies, even relatively small open-source models can match or surpass general-purpose LLMs on specialized professional tasks. However, existing approaches typically treat each domain task as a standalone or monolithic objective, training a single model to implicitly absorb multiple heterogeneous capabilities. This practice overlooks the fact that many real-world domain applications are inherently compositional.

2.2 Sparse Mixture-of-Experts

As model sizes have scaled into the hundreds of billions of parameters, sparse Mixture-of-Experts architectures have emerged as an effective way to increase model capacity via conditional computation, where a gating network dynamically routes each token to only a small subset of experts, thereby avoiding proportional growth in computational cost. (Zhang et al., 2025)

Recently, researchers have explored how to efficiently adapt MoE-based LLMs during fine-tuning by combining MoE with parameter-efficient fine-tuning techniques. LoRAMoE (Dou et al., 2024) introduces low-rank adapters as experts within an MoE framework, enabling efficient adaptation while mitigating catastrophic forgetting. MoLA (Gao et al., 2025) further extends this line of work by allocating different numbers of LoRA experts across Transformer layers to reduce redundancy and improve parameter efficiency.

Atomic Skill	Capability Description
Structure	Generating outputs that conform to predefined formats or structures.
Reasoning	Deriving conclusions through logical inference over conditions or facts.
Recall	Recalling and applying internal factual or domain-specific knowledge.
Numeracy	Performing basic numerical operations and quantitative judgments.
Expressivity	Producing open-ended, expressive, or subjective textual content.
Summarization	Condensing input text to capture core information and key points.

Table 1: Definitions of the six atomic skills used for expert decomposition.

Despite their effectiveness, these approaches primarily focus on improving parameter efficiency and expert utilization, while largely treating downstream tasks as monolithic objectives. Our work instead views domain competence as a composition of atomic skills (e.g., reasoning, numerical computation, knowledge grounding), and explores how such functional structures can be leveraged to guide expert specialization and collaboration.

3 Method

The proposed SplitThenMerge framework is organized into three successive stages, as illustrated in Figure 2. All stages are performed on top of a frozen dense LLM backbone, and no modifications are made to the base model parameters.

In the first stage, expert decomposition, domain corpora are automatically annotated at the token level with atomic skill labels using a large language model. The annotated corpus is then split and simulated to construct expert-specific training data.

In the second stage, expert training, a global

router is optimized via next-token classification on the annotated data, while multiple atomic task experts corresponding to structural, reasoning, recall, numerical, opinion, and summarization skills are trained independently using LoRA-based parameter-efficient fine-tuning.

In the final stage, expert merging, independently trained atomic experts are integrated into unified decoder layers. Each decoder layer combines base attention with model-merged LoRA attention, and a gating mechanism activates the top-1 atomic expert selected by the global router for each token. A shared LoRA feed-forward network (FFN) remains active across all tokens, and its output is combined with that of the selected atomic expert through gate-controlled weighting.

3.1 Sparse MoE Preliminaries

A transformer decoder consists of stacked decoder layers, each composed of self-attention and a feed-forward network. In sparse MoE models, the attention structure remains unchanged, while the dense FFN is replaced by a set of expert FFNs that are sparsely activated at the token level, increasing effective model capacity without modifying the standard decoding architecture.

Decoder Layer The decoder layer is the basic computational unit of autoregressive generation, sequentially transforming token representations across layers.

Attention Self-attention models contextual interactions among tokens by allowing each token to attend to previous tokens, and is typically dense and shared across all experts.

FFN The feed forward network applies nonlinear position-wise transformations independently to each token and serves as the primary source of model capacity within a decoder layer.

Expert FFN In sparse MoE models, the dense FFN is replaced by multiple expert FFNs, where only a subset of experts is activated for each token.

Gate The gate predicts token-level routing based on hidden representations, selecting which expert(s) are activated while others remain inactive.

3.2 Expert Decomposition

We begin by explicitly decomposing domain competence into a small set of atomic skills that jointly account for the functional structure of complex domain-specific generation. Based on empirical observations across domain tasks, we define six atomic skills that are both semantically distinct and practically sufficient to characterize domain outputs. Table 1 summarizes the capability scope of each skill.

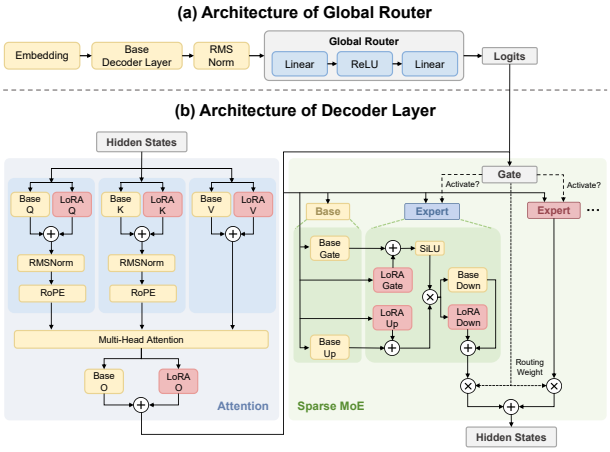

Figure 3: (a) Architecture of the global router, which predicts token-level atomic skill distributions using representations from the first decoder layer. (b) Architecture of the atomic-expert decoder layer, where base attention is combined with model-merged LoRA attention, and a sparse atomic-expert FFN is activated according to the router output.

Under this formulation, domain-specific generation is treated as a composition of these six atomic skills, with different skills activated at different stages of the output. Under this perspective, a single output sequence is no longer treated as a monolithic target, but rather as a structured mixture of heterogeneous functional behaviors.

Given a domain-specific supervised fine-tuning (SFT) dataset consisting of input–output pairs, we employ a large language model to automatically annotate each output sequence at the token level, assigning every token to exactly one atomic skill. This annotation process is fully automatic and does not rely on human labeling.

After token-level annotation, the original SFT data are split according to skill labels to construct expert-specific training corpora. For each atomic skill, we further simulate question–answer style supervision by prompting the LLM to generate skill-aligned QA pairs that reproduce the corresponding portions of the output. This procedure transforms a single monolithic output into multiple skill-specific training instances, enabling each expert to focus on a narrowly defined functional role.

3.3 Expert Training

In the second stage, we independently train a set of atomic task experts without any joint optimization

and a global router. The router focuses exclusively on token-level skill prediction, while each expert specializes in a single atomic capability through parameter-efficient fine-tuning.

Atomic Expert Training Each atomic task expert is trained independently using parameter-efficient fine-tuning. For each atomic skill, a separate LoRA adapter is inserted into the backbone model and optimized exclusively on the corresponding expert-specific training data constructed in the expert decomposition stage. This fully decoupled training strategy prevents cross-skill interference, allows each expert to specialize in a narrowly defined functional capability, and enables efficient post-hoc integration into a unified sparse MoE model.

Global Router Training The global router is trained to predict, at the token level, which atomic skill should be activated during generation. As shown in Figure 3a, the router leverages contextual representations from the frozen backbone model while avoiding any parameter modification to the base LLM. Given an input sequence, token embeddings are first obtained from the embedding layer and then forwarded through only the first decoder layer of the backbone, followed by layer normalization. This design provides sufficient contextual modeling capacity for routing while significantly reducing training cost.

Formally, let $\mathbf{h}_t \in \mathbb{R}^d$ denote the hidden state of token t produced by the frozen decoder layer. A shallow classifier head maps $\mathbf{h}_t$ to a categorical distribution over K atomic skills:

$$\mathbf{p}_t = \mathrm{Softmax}(W_2\, \sigma(W_1 \mathbf{h}_t)), \tag{1}$$

where W_1 and W_2 are trainable parameters and $\sigma(\cdot)$ denotes the ReLU activation. The router is optimized using token-level supervision obtained from automatic atomic annotation. The training objective is the cross-entropy loss:

$$\mathcal{L}_{\text{router}} = -\sum_{t=1}^{T} \log p_t(y_t), \tag{2}$$

where y_t denotes the ground-truth atomic skill label of token t, and non-target tokens are masked out during optimization.

3.4 Expert Merging

Finally, independently trained experts are merged into a unified sparse MoE model.

3.4.1 Decoder Architecture

As shown in Figure 3b, each decoder layer integrates base attention, merged LoRA attention, and the sparse atomic-expert FFN with global routing.

Attention Let $\mathbf{x}_t \in \mathbb{R}^d$ be the hidden state at token position t. For self-attention, we use base projections W_Q, W_K, W_V, W_O and a LoRA update ΔW_*:

$$\begin{aligned} \mathbf{q}_t &= (W_Q + \Delta W_Q)\mathbf{x}_t, \\ \mathbf{k}_t &= (W_K + \Delta W_K)\mathbf{x}_t, \\ \mathbf{v}_t &= (W_V + \Delta W_V)\mathbf{x}_t, \\ \mathbf{o}_t &= (W_O + \Delta W_O)\mathbf{z}_t, \end{aligned} \tag{3}$$

where $\mathbf{z}_t$ denotes the attention output before the O projection.

Routing The routing signal is computed once per forward pass by the global router after the first decoder layer, producing token-wise group logits:

$$\mathbf{r}_t = \mathrm{GlobalRouter}(\mathrm{Norm}(\mathbf{h}_t)) \in \mathbb{R}^G, \tag{4}$$

where G is the number of atomic experts and $\mathbf{h}_t$ denotes the hidden state at that layer. The activated atomic expert is determined solely by the router via top-1 selection:

$$e_t = \arg\max_{e \in \{1,\ldots,G\}} \mathbf{r}_t[e]. \tag{5}$$

Within each decoder layer, a local gate produces mixture weights over all FFN branches, including one always-active shared expert e_0 and the routed atomic experts:

$$\begin{aligned} \mathbf{s}_t &= W_{\text{gate}}\mathbf{x}_t \in \mathbb{R}^{(G+1)}, \\ \boldsymbol{\pi}_t &= \mathrm{Softmax}(\mathbf{s}_t). \end{aligned} \tag{6}$$

Given the router-selected expert e_t, we form the active set $\mathcal{S}_t = \{e_0, e_t\}$ and take the corresponding gate probabilities as raw routing weights:

$$\tilde{w}_{t,e} = \boldsymbol{\pi}_t[e], \quad e \in \mathcal{S}_t. \tag{7}$$

We renormalize them over the active set:

$$w_{t,e} = \frac{\tilde{w}_{t,e}}{\sum_{e' \in \mathcal{S}_t} \tilde{w}_{t,e'}}, \quad e \in \mathcal{S}_t. \tag{8}$$

Finally, the FFN output is the weighted sum of the always-on shared branch and the selected atomic expert:

$$\mathbf{y}_t = \sum_{e \in \mathcal{S}_t} w_{t,e} \cdot \mathrm{FFN}^{(e)}(\mathbf{x}_t). \tag{9}$$

Therefore, the global router determines which atomic expert is activated, while the local gate provides the mixture weights used to combine it with the always-active expert.

Expert FFN Each FFN branch adopts a LoRA-augmented gated MLP with gate, up and down projections. Let W_g, W_u, W_d denote the base FFN projections and let $\Delta W_g^{(e)}, \Delta W_u^{(e)}, \Delta W_d^{(e)}$ denote the LoRA updates of expert e.

The expert-specific FFN is computed as:

$$\mathrm{FFN}^{(e)}(\mathbf{x}_t) = \big(W_d + \Delta W_d^{(e)}\big)\Big(\phi\big((W_g + \Delta W_g^{(e)})\,\mathbf{x}_t\big) \odot \big((W_u + \Delta W_u^{(e)})\mathbf{x}_t\big)\Big), \tag{10}$$

where $\phi(\cdot)$ denotes the activation and $\odot$ denotes element-wise multiplication. The always-on shared expert e_0 follows the same formulation with its own LoRA parameters and is always included in the active set $\mathcal{S}_t$ in Eq. 9.

3.4.2 Merging Strategy

After independent training, we integrate atomic experts into a unified decoder by separately handling attention and FFN components.

Attention Construction For each decoder layer and each attention projection $* \in \{Q, K, V, O\}$, every atomic expert e provides a LoRA pair $(A_*^{(e)}, B_*^{(e)})$. Instead of merging the induced update $\Delta W_*^{(e)} = B_*^{(e)} A_*^{(e)}$, we directly merge the low-rank factors. Concretely, we apply Dare-TIES merging (Yu et al., 2024) to the sets of A-matrices and B-matrices separately:

$$\begin{aligned} \bar{A}_* &= \mathrm{DareTIES}\Big(\{A_*^{(e)}\}_{e=1}^{G}; \rho\Big), \\ \bar{B}_* &= \mathrm{DareTIES}\Big(\{B_*^{(e)}\}_{e=1}^{G}; \rho\Big), \end{aligned} \tag{11}$$

where ρ denotes the density hyperparameter in Dare-TIES. The merged attention adapter is then parameterized as $(\bar{A}_*, \bar{B}_*)$ and used in the standard LoRA form $\Delta W_* = \bar{B}_* \bar{A}_*$.

FFN Construction In contrast, FFN experts are not merged in parameter space. For each layer, we copy the backbone FFN projections into the merged model and directly load each atomic expert's LoRA parameters into its corresponding expert slot. This copy-based construction preserves expert-specific transformations and enables router-controlled sparse activation.

4 Experiments

We evaluate the proposed SplitThenMerge framework on two complex domain-specific generation tasks from the legal and medical domains.

4.1 Experimental Setup

We report the main implementation details of our experiments. Full settings and hyperparameters are provided in Appendix A.

4.1.1 Court View Generation

Court view generation aims to produce the judicial reasoning section of a judgment based on the factual description of a case (Li et al., 2024). For criminal cases, the generated court view typically includes factual findings, legal analysis, and sentencing decisions. We collect 10,056 Chinese criminal judgments from China Judgments Online[1] platforms. Among them, 9,956 documents are used for training and 1,000 for testing.

We adopt Qwen3-8B (Yang et al., 2025) as the backbone model. Following prior work in court view generation, we evaluate model performance using both paragraph-level similarity metrics and task-specific correctness metrics. Specifically, we report BLEU-1 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) to measure textual similarity, mean absolute error (MAE) in months for sentencing term prediction, as well as sample-level F1 scores for charge prediction and law article prediction.

4.1.2 Discharge Summary Generation

Discharge summary generation is a medical text generation task that aims to produce discharge summaries based on a patient's admission information and hospitalization records (Xu et al., 2024). The generated summaries include diagnoses, prescribed medications, medication dosages, and post-discharge instructions. We construct the dataset from the MIMIC-III database (Johnson et al., 2016), which integrates deidentified clinical data collected during routine hospital care at the Beth Israel Deaconess Medical Center in Boston. The dataset contains a total of 18,514 instances, with

[1] https://wenshu.court.gov.cn/

Model	Method	Params	Val Loss ↓	BLEU-1	ROUGE-L	Sentence ↓	Article	Charge
Qwen3-8B	Base	–	/	49.22	14.44	4.85	50.84	96.25
	Multi-Agent	–	/	43.76	14.88	3.89	60.26	96.52
	LoRA (r=16)	42M	0.8235	49.75	14.92	4.57	52.29	96.36
	LoRA (r=128)	339M	0.5591	56.09	22.06	4.26	74.27	95.67
	Ours (w/o merging)	225M	0.5242	66.91	28.62	3.54	75.73	96.19
	Ours (w merging)	208M	**0.4727**	**68.49**	**31.63**	**3.41**	**78.26**	**96.82**

Table 2: Performance on the task of court view generation with Qwen3-8B. The best results highlighted in **bold**.

Model	Method	Params	Val Loss ↓	BLEU-1	ROUGE-L	Disease	Medication	Dosage
Llama-3.1-8B	Base	–	/	18.40	7.35	31.43	36.44	23.95
	Multi-Agent	–	/	19.05	9.97	39.08	40.52	27.59
	LoRA (r=16)	40M	1.2664	22.69	11.85	43.03	37.30	20.81
	LoRA (r=128)	325M	0.9809	23.55	18.62	46.76	43.84	27.71
	Ours (w/o merging)	222M	0.8018	41.18	31.33	48.20	44.13	27.94
	Ours (w merging)	206M	**0.6452**	**42.74**	**33.65**	**50.94**	**52.44**	**36.85**

Table 3: Performance on the task of discharge note generation with Llama-3.1-8B.

17,514 used for training and 1,000 reserved for testing.

All experiments are conducted using Llama-3.1-8B-Instruct (Dubey et al., 2024) as the backbone model. Generation quality is evaluated using BLEU-1 and ROUGE-L, and medical factual correctness is assessed using sample-level F1 scores for disease diagnosis prediction, medication name prediction, and medication dosage prediction.

4.1.3 Baselines

We compare against a zero-training **Multi-Agent** prompting baseline (Tran et al., 2025), where different agents specialize in different skills and their outputs are heuristically combined, with the implementation detailed in Appendix B, as well as **LoRA** fine-tuning (Hu et al., 2022), which performs standard parameter-efficient adaptation using LoRA adapters with rank configurations matched to our parameter budget.

4.1.4 Implementation Details

Each atomic expert is independently trained for one epoch on its corresponding skill-specific data using LoRA fine-tuning. After expert training, the experts are integrated into a unified sparse MoE model following the proposed merging strategy. The merged MoE model is then lightly fine-tuned for only 10 steps on each task, while all original backbone model parameters remain frozen.

4.2 Main Results

Tables 2 and 3 present the main results for court view generation in the legal domain and discharge summary generation in the medical domain, respectively. Across both tasks, SplitThenMerge consistently achieves the best overall performance under comparable or even smaller trainable parameter budgets, indicating the effectiveness of explicitly decomposing domain competence into atomic skills and coordinating them at the token level.

On the legal court view generation task based on Qwen3-8B, SplitThenMerge significantly outperforms the base model, multi-agent prompting, and standard LoRA fine-tuning. Compared with the base model, the merged SplitThenMerge model improves ROUGE-L from 14.44 to 31.63 and reduces sentencing error, with MAE decreasing from 4.85 to 3.41 months. It also yields substantial gains on task-specific correctness metrics, with Article F1 increasing from 50.84 to 78.26.

Notably, these improvements are not merely a result of increased model capacity. Even compared with a much larger LoRA configuration (339M trainable parameters), SplitThenMerge uses fewer parameters (208M) while achieving stronger overall performance, highlighting its superior parameter efficiency.

A similar trend is observed in the medical discharge summary generation task using Llama-3.1-8B-Instruct. The merged SplitThenMerge model yields over 20% improvement in semantic similarity metrics and more than 15% gains on medical factual accuracy for diseases, medications, and dosages. Compared with larger LoRA baselines, SplitThenMerge still achieves average improvements of 17% on similarity metrics and 7% on

Domain	Model	Params	Val Loss ↓	BLEU-1	ROUGE-L	Sentence ↓	Article	Charge
Law	SplitThenMerge Decoder	225M	0.5242	66.91	28.62	3.54	75.73	96.19
	+ Global Router	208M	0.5230	66.86	29.75	3.61	75.33	96.33
	+ Merging FFN	208M	0.4761	68.31	31.04	3.49	78.17	96.53
	+ Merging Attention	208M	**0.4727**	**68.49**	**31.63**	**3.41**	**78.26**	**96.82**
Domain	**Model**	**Params**	**Val Loss ↓**	**BLEU-1**	**ROUGE-L**	**Disease**	**Medication**	**Dosage**
Medical	SplitThenMerge Decoder	222M	0.8018	41.18	31.33	48.20	44.13	27.94
	+ Global Router	206M	0.7984	38.41	29.50	50.08	42.93	26.01
	+ Merging FFN	206M	0.6630	42.40	33.05	49.08	50.54	35.20
	+ Merging Attention	206M	**0.6452**	**42.74**	**33.65**	**50.94**	**52.44**	**36.85**

Table 4: Ablation study on legal judgment generation and discharge note generation, where we progressively add the global router and expert merging components.

factual correctness.

Furthermore, the comparison with multi-agent prompting provides insight into the granularity of skill coordination. While multi-agent approaches consistently outperform the base model, confirming the benefit of explicit skill decomposition, they remain inferior to SplitThenMerge. This result suggests that prompt-level agent collaboration alone is insufficient for fine-grained skill coordination, whereas parameter-level skill specialization combined with token-level routing leads to more effective integration of heterogeneous domain skills.

4.3 Ablation Study

We conduct an ablation study to analyze the contribution of each component in SplitThenMerge. As shown in Table 4, we progressively introduce the global router and expert merging modules on top of the SplitThenMerge decoder. Across both domains, the full model consistently achieves the best overall performance, indicating that solving complex domain-specific tasks requires not only token-level skill-aware routing, but also the effective integration of independently trained experts into the decoding process.

Further analysis shows that merging independently trained experts is critical for performance improvements. In particular, incorporating expert feed-forward networks into the decoder leads to substantial reductions in validation loss across both domains, which is consistent with the common understanding that domain knowledge and task-specific transformations are primarily encoded in feed-forward layers. In contrast, introducing the global router alone, without merging the corresponding experts, yields little to no performance gain. This indicates that routing signals are only effective when coupled with concrete expert representations to activate.

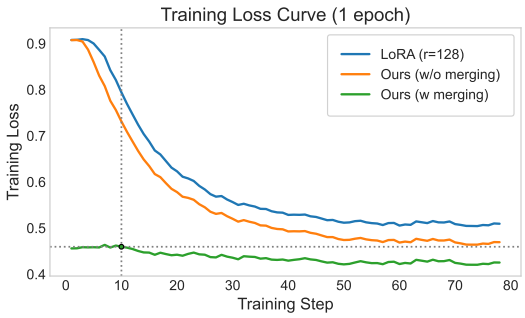

Figure 4: Training loss curves on the legal domain.

4.4 Training Convergence Analysis

Figure 4 shows that the merged SplitThenMerge model converges substantially faster on the legal domain. It starts from a lower initial loss, while the LoRA and the non-merged variant require nearly a full training epoch to reach comparable loss levels. This validates the effectiveness of expert merging, which yields a better-initialized and more coordinated model and facilitates more efficient optimization.

5 Concolution

This paper views complex domain-specific tasks as compositions of atomic skills and explicitly incorporates such skills into domain-adapted model training. We propose SplitThenMerge, a skill-oriented framework that performs expert decomposition, independent expert and global-router training, and expert merging, enabling atomic experts to achieve effective specialization and frequent fine-grained coordination within a unified token-level sparse MoE model. Experiments on legal and medical generation tasks demonstrate that, under comparable or even smaller training parameter budgets, SplitThenMerge consistently achieves superior performance.

Limitations

This paper has two potential limitations. First, SplitThenMerge assumes a predefined set of atomic skills that is manually designed based on observations. While the six-skill taxonomy covers common functional patterns in specific domains, it may not be optimal or complete for all domains. This limitation lies in the current implementation rather than the framework itself, which is inherently flexible and can accommodate a larger or different set of skills.

Second, the current implementation assumes that each token activates a single atomic skill expert. The key contribution of this work lies in the skill-oriented perspective and the unified framework for token-level expert coordination, which can be naturally extended to top-k or soft expert activation schemes when greater expressiveness is required.

Ethics Statement

Legal and medical artificial intelligence systems operate in highly sensitive domains, where errors or hallucinations may lead to serious real-world consequences. Precisely because of this sensitivity, it is crucial to improve model performance, reliability, and domain-specific accuracy in these settings. The datasets used in this work, including judicial documents and medical records, are obtained from publicly available sources and have been de-identified to remove any personally identifiable information.

Acknowledgments

This work was supported in part by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037, 2024C01259), National Key Research and Development Program of China (2024YFE0203700), and National Natural Science Foundation of China (62376243), Key R&D Program of Hangzhou (2025SZDA0254), Ant Group, Chongqing Ant Consumer Finance Co., Ant Group through CCF-Ant Research Fund. All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Hallucinating law: Legal mistakes with large language models are pervasive. *Law, regulation, and policy*.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Ruibo Liu, Baochen Sun, Yawen Zhang, Daiyi Peng, Xiaoyuan Guo, and Vs Subrahmanian. 2025. MoLA: MoE LoRA with layer-wise expert allocation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5097–5112, Albuquerque, New Mexico. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Rebecca Handler, Sonali Sharma, and Tina Hernandez-Boussard. 2025. The fragile intelligence of gpt-5 in medicine. *Nature Medicine*, pages 1–3.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Alistair EW 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(1):1–9.

Ang Li, Yiquan Wu, Yifei Liu, Kun Kuang, Fei Wu, and Ming Cai. 2024. Enhancing court view generation with knowledge injection and guidance. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5896–5906, Torino, Italia. ELRA and ICCL.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6).

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

OpenAI. 2025. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Justin Xu, Zhihong Chen, Andrew Johnston, Louis Blankemeier, Maya Varma, Jason Hom, William J. Collins, Ankit Modi, Robert Lloyd, Benjamin Hopkins, Curtis Langlotz, and Jean-Benoit Delbrouck. 2024. Overview of the first shared task on clinical text generation: RRG24 and "discharge me!". In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 85–98, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, et al. 2024. Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval. In *International Conference on Database Systems for Advanced Applications*, pages 304–321. Springer.

Danyang Zhang, Junhao Song, Ziqian Bi, Yingfang Yuan, Tianyang Wang, Joe Yeong, and Junfeng Hao. 2025. Mixture of experts in large language models. *arXiv preprint arXiv:2507.11181*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Experiment Settings

This appendix provides implementation details, including model configurations, hyper-parameters, and training settings used in our experiments.

A.1 Settings of LLMs

All large language models use the default system prompt: `"You are a helpful assistant."`

We adopt the default generation configurations provided by each model. For Qwen3-8B, we set the temperature to 0.6 and top-p to 0.95. For LLaMA-3.1-8B-Instruct, we use a temperature of 0.6 and top-p of 0.9.

A.2 Hyper-parameters of SplitThenMerge

All atomic experts are implemented using LoRA-based parameter-efficient fine-tuning. Each expert uses a LoRA rank of 16 and a LoRA scaling factor of 32. In total, we instantiate seven experts, consisting of one always-active shared expert and six atomic skill experts corresponding to the predefined abilities.

During attention construction, we apply Dare-TIES merging with a drop rate of 50% for merging LoRA attention adapters.

A.3 Settings of Supervised Fine-tuning

We conduct supervised fine-tuning using the llama-factory framework (Zheng et al., 2024). All experiments are trained on 8 NVIDIA H100 GPUs with DeepSpeed ZeRO-0 optimization. The AdamW optimizer is used throughout training and the global training batch size is set to 128.

For the global router, we set the learning rate to 2×10^{-4} and train for 10 epochs on next token classification. For the atomic skill experts, each expert is trained independently using a learning rate of 1×10^{-5} for a single epoch on its corresponding skill-specific dataset. During the final expert merging stage, the merged sparse MoE model is further fine-tuned for 10 update steps with a learning rate of 5×10^{-5}, while all backbone model parameters remain frozen.

A.4 Prompt for Token-level Atomic Expert Annotation

We use the following prompt to automatically annotate outputs at a fine-grained, token-level resolution with atomic skill labels.

Please annotate the output by identifying which atomic skill is required to produce each part of the output given the input.

The output should be segmented according to semantic boundaries, and each segment should be assigned exactly one skill.

Skills

Structure: Generating outputs that conform to predefined formats or structures.

Reasoning: Deriving conclusions through logical inference over conditions or facts.

Recall: Recalling and applying internal factual or domain-specific knowledge.

Numeracy: Performing basic numerical operations and quantitative judgments.

Expressivity: Producing open-ended, expressive, or subjective textual content.

Summarization: Condensing input text to capture core information and key points.

Example

Below is an illustrative example.

Input

{example input}

Output

{example output}

Annotation

{example annotation}

Reason

{example reason}

Task

Now you are given a real task to process.

Input

{input}

Output

{output}

Annotation

Output a JSON list.

Each semantically coherent sentence or segment in the output must be annotated with one atomic skill.

Reason

In this section, explain the reason for your annotations.

A.5 Prompt for Expert Training Data Construction

After token-level atomic annotation, we further construct expert-specific training data by prompting a large language model to generate skill-aligned question–answer pairs.

The following prompt is used to ensure that each question exclusively targets a single atomic skill and that the corresponding answer is grounded in a specified text segment.

> Given the following text, construct a question.
> The goal of the question is to assess the {skill} capability, defined as: {description}.
> The source of the answer is explicitly specified and must be strictly followed.
> # Text
> {instruction}
> {input}
> # Answer Source
> {segment}
> # Question
> Describe the question you construct here.
> # Answer
> Provide a coherent and complete answer here. The answer must be derived exclusively from the given answer source and must not include any additional information.

B Prompts for Multi-Agent Baseline

The multi-agent approach does not involve any parameter updates and relies solely on role-based prompting to simulate skill specialization. Each agent is assigned a fixed role corresponding to an atomic skill, and all agents operate on the same input independently. Their outputs are then heuristically aggregated to form the final response.

B.1 Prompts for Agent

We instantiate six role-based agents corresponding to the atomic skills in Table 1, each executing the following prompt with its own skill description.

> You are a domain expert specialized in {skill}.
> From the perspective of {skill} ({skill description}), analyze and identify the information required to complete the task: {task instruction}.
> Input: {task input}

B.2 Prompts for Aggregation

After all agents independently produce skill-specific analyses, a final aggregation model is prompted to integrate their outputs and generate the task output.

> Please integrate the outputs from all agents to complete the following task.
> ## Agent Outputs
> ### {skill}
> {agent output}
> ### {skill}
> {agent output}
> ...
> ## Task
> {introduction}
> {input}

C Case Study

Figure 5 illustrates a representative example of discharge summary generation for a patient admitted with hyponatremia and multiple comorbidities. Compared with standard LoRA fine-tuning, SplitThenMerge produces more complete discharge medications and diagnoses with higher clinical consistency. The LoRA baseline omits several important medications and exhibits mismatches between diagnoses and prescribed treatments. After expert merging, the final SplitThenMerge model further improves precision and coherence, yielding a discharge summary that more closely resembles real-world clinical documentation. This improvement reflects more effective token-level coordination among atomic experts, enabling the model to dynamically switch between knowledge recall, structured listing, and numerical dosage specification during generation.

Overall, this case study qualitatively demonstrates that token-level expert coordination leads to more accurate and reliable domain-specific generation than monolithic fine-tuning.

Admission Date: [PHI] Discharge Date: [PHI] Date of Birth: [PHI] Sex: F Service: MEDICINE Allergies: Patient recorded as having No Known Allergies to Drugs Attending: [PHI] Chief Complaint: hyponatremia Major Surgical or Invasive Procedure: none History of Present Illness: 83-year-old woman with history of CAD (s/p CABGx2 ' [PHI]) and DM2 presented to [PHI] with confusion x 1 week and weakness x 4 weeks. The patient has experienced weakness with an 8-lb weight loss since [PHI] . Three weeks ago she had a few episodes of nonbloody diarrhea, presented to [PHI] a few weeks ago for work-up, which was reportedly unrevealing. She was then diagnosed with an asymptom UTI, treated with antibiotics, during that admission. The diarrhea resolved after the discontinuation of stool softeners and she was discharged home. For the past week, according to her son, she was confused intermittently. She reports having poor PO intake for the past few weeks. Patient talked to her PCP, [PHI] . [PHI] , on the phone the day of admission and reportedly had some confused speech. She presented to [PHI] ED and was found to have Na 110 and transferred to [PHI] after getting 250 ml of NS then 3% NaCl IVF at 29 cc/hr. . In ED, T 98.0, BP 156/67, HR 66, RR 20, O2 sat 99%. Renal was consulted and recommended 3% NaCl at 15 ml/hr with q4h Na checks

Label	LoRA (r=128)	Ours (w/o merging)	Ours (w merging)
Discharge Medications: 1. Trazodone 50 mg Tablet Sig: 0.5 Tablet PO HS (at bedtime) as needed. 2. Pantoprazole 40 mg Tablet, Delayed Release (E.C.) Sig: One (1) Tablet, Delayed Release (E.C.) PO Q24H (every 24 hours). 3. Aspirin 81 mg Tablet, Chewable Sig: One (1) Tablet, Chewable PO DAILY (Daily). 4. Atorvastatin 20 mg Tablet Sig: One (1) Tablet PO DAILY (Daily). 5. Carvedilol 6.25 mg Tablet Sig: One (1) Tablet PO BID (2 times a day). 6. Valsartan 160 mg Tablet Sig: One (1) Tablet PO DAILY (Daily). 7. Glyburide 5 mg Tablet Sig: One (1) Tablet PO BID (2 times a day). 8. Alendronate 70 mg Tablet Sig: One (1) Tablet PO QMON (every Monday). 9. Calcium Carbonate 500 mg Tablet, Chewable Sig: One (1) Tablet, Chewable PO TID (3 times a day). 10. Cholecalciferol (Vitamin D3) 400 unit Tablet Sig: One (1) Tablet PO DAILY (Daily). 11. Multivitamin Tablet Sig: One (1) Tablet PO DAILY (Daily).	Discharge Medications: 1. Aspirin 325 MG Tablet Sig: One (1) Tablet PO DAILY (Daily). Disp:*90 Tablet(s)* Refills:*2* 2. Atorvastatin 20 MG Tablet Sig: One (1) Tablet PO Daily (Daily). Disp:*30 Tablet(s)* Refills:*2* 3. Calcium Carbonate 500 MG Tablet Sig: Two (2) Tablet PO HS (at bedtime). Disp:*60 Tablet(s)* Refills:*2* 4. Cholecalciferol 400 Unit Capsule Sig: One (1) Capsule PO DAILY (Daily). Disp:*30 Capsule(s)* Refills:*2* 5. Ferrous Sulfate 300 MG Tablet Sig: One (1) Tablet PO DAILY (Daily). Disp:*30 Tablet(s)* Refills:*2* 6. Omeprazole 20 MG Capsule, Delayed Release(Sustained Release), ER Sig: One (1) Capsule, Delayed Release(Sustained Release), ER PO DAILY (Daily). Disp:*30 Capsule, Delayed Release(Sustained Release), ER(s)* Refills:*2* 7. Carvedilol 6.25 MG Tablet Sig: One (1) Tablet PO BID (2 times a day). Disp:*60 Tablet(s)* Refills:*2* 8. Ezetimibe 10 MG Tablet Sig: One (1) Tablet PO DAILY (Daily). Disp:*30 Tablet(s)* Refills:*2* 9. Glyburide 1.25 MG Tablet Sig: One (1) Tablet PO BID (2 times a day). Disp:*60 Tablet(s)* Refills:*2* 10. Hydrochlorothiazide 12.5 MG Tablet Sig: One (1) Tablet PO DAILY (Daily). Disp:*30 Tablet(s)* Refills:*2* 11. Simvastatin 40 MG Tablet Sig: One (1) Tablet PO DAILY (Daily). Disp:*30 Tablet(s)* Refills:*2* Discharge Disposition: Extended Care Facility: [PHI]	Discharge Medications: 1. Aspirin 325 mg Tablet, Delayed Release (E.C.) Sig: One (1) Tablet, Delayed Release (E.C.) PO DAILY (Daily). 2. Atorvastatin 20 mg Tablet Sig: One (1) Tablet PO DAILY (Daily). 3. Carvedilol 3.125 mg Tablet Sig: Two (2) Tablet PO twice a day. 4. Ezetimibe 10 mg Tablet Sig: One (1) Tablet PO once a day. 5. Ferrous Sulfate 325 (65) mg Tablet Sig: One (1) Tablet PO QHS (once a day (at bedtime)). 6. Metoprolol Tartrate 50 mg Tablet Sig: One (1) Tablet PO three times a day. 7. Multivitamin Tablet Sig: One (1) Tablet PO once a day. 8. Pantoprazole 40 mg Tablet, Delayed Release (E.C.) Sig: One (1) Tablet, Delayed Release (E.C.) PO Q24H (every 24 hours). 9. Valsartan 160 mg Tablet Sig: One (1) Tablet PO twice a day. 10. Oxycodone-Acetaminophen 5-325 mg Tablet Sig: One (1) Tablet PO every 4-6 hours as needed for pain. 11. Insulin Regular Human 100 unit/mL Solution Sig: Subcutaneous as needed for blood glucose control. 12. Glyburide 5 mg Tablet Sig: One (1) Tablet PO twice a day.	Discharge Medications: 1. alendronate 70 mg Tablet Sig: One (1) Tablet PO QWednesdays. 2. Aspirin 325 mg Tablet, Chewable Sig: One (1) Tablet, Chewable PO DAILY (Daily). 3. Calcium Carbonate-Vitamin D3 600-200 mg-unit Capsule, Chewable Sig: One (1) Cap Chewable PO once a day. 4. Multivitamin Capsule Sig: One (1) Cap PO once a day. 5. Atorvastatin 20 mg Tablet Sig: One (1) Tablet PO DAILY (Daily). 6. Iron 325 (65) mg Tablet Sig: One (1) Tablet PO once a day. 7. Ezetimibe 10 mg Tablet Sig: One (1) Tablet PO DAILY (Daily). 8. Pantoprazole 40 mg Tablet, Delayed Release (E.C.) Sig: One (1) Tablet, Delayed Release (E.C.) PO Q12H (every 12 hours). 9. Carvedilol 6.25 mg Tablet Sig: Two (2) Tablet PO twice a day. 10. Valsartan 160 mg Tablet Sig: Two (2) Tablet PO once a day. 11. Glyburide 5 mg Tablet Sig: One (1) Tablet PO twice a day.
Discharge Diagnosis: Hyponatremia Depression	Discharge Diagnosis: 1. Hyponatremia 2. Confusion 3. Depression 4. Diabetes Mellitus Type II 5. Coronary Artery Disease Status Post Coronary Artery Bypass Graft 6. Hypertension	Discharge Diagnosis: Primary: Hyponatremia Confusion Secondary: CAD Depression DM2	Discharge Diagnosis: Hyponatremia Depression
Discharge Disposition: Extended Care Facility: [PHI] Healthcare Center Discharge Condition: good Discharge Instructions: stop metformin because of low appetite and diarrhea. Fluid restriction of 1200 ML daily. Stop Celexa monitor Sodium level twice weekly for 2 weeks and then, if levels are stable, once weekly for 1 month. Followup Instructions: PCP: [PHI] , [PHI] C. [PHI] . Follow up with geriatrics and/or neuropsychiartry.	Discharge Condition: Alert and interactive. Discharge Instructions: You were admitted to the hospital for severe hyponatremia. Your sodium levels have returned to normal while you were in the hospital. However, you still have confusion and difficulty walking. These symptoms should improve over time but it is not clear why they started. Please call your primary care doctor's office to schedule an appointment to discuss this further. If you experience any worsening of these symptoms please seek immediate attention. You will need to followup with the rehabilitation center where you will receive physical therapy, occupational therapy and speech therapy. Please followup with your primary care provider as scheduled. Followup Instructions: Primary Care Provider: [PHI], MD Phone: [PHI] Office Hours: [PHI] [PHI] :00am [PHI] :00pm Rehabilitation Center: [PHI] [PHI] [PHI] : [PHI] Phone: [PHI] Date/Time: [PHI] [PHI] : [PHI] Time: [PHI] [PHI] : [PHI] : [PHI], MD Phone: [PHI] Office Hours: [PHI] [PHI] :00am [PHI] :00pm	Discharge Disposition: Extended Care Facility: [PHI] - [PHI] Discharge Condition: mental status is clear and coherent. Level of consciousness is alert and interactive. Activity status is ambulatory - independent. Discharge Instructions: Please take your medications as prescribed. You should drink enough fluids to keep urine light yellow in color. If you develop shortness of breath, chest pain, increased confusion, or other concerning symptoms please call your doctor or go to the emergency room. Followup Instructions: Department: NEUROPSYCHIATRY When: THURSDAY [PHI] at 9:00 AM With: [PHI] [PHI], MD [PHI] Building: SC [PHI] Clinical Ctr [PHI] Campus: EAST Best Parking: [PHI] Garage Department: NEUROLOGY When: FRIDAY [PHI] at 9:30 AM With: [PHI] [PHI], MD [PHI] Building: SC [PHI] Clinical Ctr [PHI] Campus: EAST Best Parking: [PHI] Garage Completed by: [PHI]	Discharge Disposition: Extended Care Facility: [PHI] HealthCare & Rehabilitation Center - [PHI] Discharge Condition: Mental Status: Clear and coherent. Level of Consciousness: Alert and interactive. Activity Status: Ambulatory - Independent. Discharge Instructions: You were admitted to the hospital because you were confused, weak, and had low sodium levels. Your sodium levels returned to normal while you were here. You should drink less than one cup of fluid per meal and not more than two cups throughout the day. If you experience any other concerning symptoms such as confusion, dizziness, shortness of breath, chest pain, or severe headaches please call your doctor or return to the emergency room. Followup Instructions: Please follow up with Dr. [PHI] [PHI] [PHI] [PHI] in 2 weeks. Completed by: [PHI]

Figure 5: Comparison of discharge medications, diagnoses, and instructions generated by LoRA, SplitThenMerge without merging, and SplitThenMerge with expert merging, highlighting improved medication accuracy.