CO-OPTIMIZING RECOMMENDATION AND EVALUATION FOR LLM SELECTION

Tarun Kumar[1] **Cong Xu**[2] **Arpit Shah**[1] **Baradji Diallo**[3] **Martin Foltin**[3]
Suparna Bhattacharya[1]
[1]Hewlett Packard Labs, Bangalore, India [2]Hewlett Packard Labs, Singapore
[3]Hewlett Packard Labs, USA
{tarun.kumar2, cong.xu, arpit.shah, baradji.diallo, martin.foltin, suparna.bhattacharya}@hpe.com

ABSTRACT

The rapid expansion of Large Language Models (LLMs) introduces a new conundrum for AI deployments: efficiently selecting the most appropriate model for a given real-world task from a long tail of specialized models and tasks which are underrepresented in popular leaderboards. Recent advances in LLM routing methods enable fine-grained selection by mapping prompts to an optimal model from a limited pool. However, identifying this small model pool remains a non-trivial challenge, with over 180,000 public LLMs available, nearly 10,000 new models emerging each month, and the mounting computational cost of comprehensive evaluations. We introduce RELM (Recommender Engine for Large Models), a scalable framework designed to identify the most suitable LLMs for specific user tasks. RELM selects benchmarks to evaluate LLMs using a companion multistage evaluation framework, HERD (Holistic Evaluation, Ranking, and Deciphering). Co-optimization of RELM and HERD balances the need for evaluating new models while learning from existing evaluations. Our results demonstrate that RELM-recommended models outperform HuggingFace search recommended models and other popular models in open-ended text generation and LLM-based classification tasks in the Healthcare, chemistry, and finance domains. Further, when integrated with an existing LLM routing system, RELM results in performance gains of 54.5% and 175% for ROUGE-L and BLEU scores respectively.

1 INTRODUCTION

The rapid advancement of large language models (LLMs) has revolutionized natural language processing (NLP), achieving state-of-the-art performance across diverse tasks such as text generation, summarization, and reasoning Asai et al. (2024); Dekoninck et al. (2024), Liu et al. (2024), Yao et al. (2024); Hao et al. (2023).However, the exponential growth in the number and diversity Naveed et al. (2023) of LLMs presents a critical challenge: selecting the most suitable model for a given task. With over 180,000 public LLMs and nearly 10,000[1] new ones emerging each month, manual selection is infeasible. While curated leaderboards exist, they often prioritize general-purpose models, overlooking specialized or emerging models better suited for domain-specific applications. Additionally, many public LLMs lack comprehensive benchmark results, making it difficult to assess their quality. Selecting the optimal model requires balancing multiple factors, including task relevance, domain expertise, performance constraints, and computational resources. This challenge is particularly acute in fields like healthcare and chemistry, where general-purpose models may lack specialized knowledge, while fine-tuned models often lack broad benchmark coverage. Existing LLM routing methods attempt to address this problem by mapping user prompts to models from a predefined small pool, yet they do not answer a more fundamental question: how do we determine which models should be in that pool? A scalable and intelligent approach is needed to efficiently identify, evaluate, and recommend LLMs without requiring exhaustive benchmarking.

At the core of this challenge lies a fundamental chicken-and-egg problem: benchmark results are needed to make informed model recommendations, yet evaluations are computationally expensive

[1]URL: `https://huggingface.co/models` accessed on Feb 9, 2025

and must be selectively prioritized. The solution must: (a) discover promising models from the vast model space while minimizing evaluation costs; (b) identify relevant benchmarks that truly reflect real-world performance requirements; (c) continuously incorporate new evaluation results to improve future recommendations.

To address this, we propose RELM (Recommender Engine for Large Models), a scalable system that recommends the most suitable LLMs and benchmarks while minimizing evaluation costs. RELM integrates with HERD (Holistic Evaluation, Ranking, and Deciphering), a multistage evaluation framework that balances the need for exploration (discovering new models) with exploitation (leveraging existing evaluations). **We make the following contributions**:
(1) Metadata-Driven Foundation Model Recommendation System: We introduce RELM, a scalable recommendation system leveraging metadata characteristics (such as application domain, task details, etc.) to suggest potential <LLM, benchmark> pairs for evaluation, identifying their fitness for specific use cases.
(2) Novel Co-optimization Framework for Model Selection and Evaluation: We present a novel system that jointly optimizes model recommendations and benchmark evaluations through an iterative feedback loop. HERD employs a multistage evaluation process, efficiently filtering less suitable pairs thus reducing costs and improving recommendations.
(3) Hierarchical Model Selection for Enhanced LLM Routing: We introduce a novel two-tier approach that bridges the gap between large-scale model discovery via RELM-HERD co-optimization and fine-grained LLM routing Guha et al. (2024); Ong et al. (2024).

In the remainder of this paper: Section 2 discusses related work and positions our research within the existing literature. Section 3 formulates our research problem, followed by a detailed presentation of our proposed method, RELM, HERD and their co-optimization in Section 4. We describe our experimental setup in Section 5 and provide a comprehensive analysis of results in Section 6.

2 Related work

The challenge of selecting optimal models and other ML-pipeline artifacts for specific tasks has been a long-standing problem in machine learning, predating the era of large language models (LLMs) Mazaheri et al. (2021); Green (2023); Dukhanov et al. (2024). Traditional recommender systems have played a crucial role in model selection, employing techniques such as collaborative filtering Yang et al. (2019), content-based methods Cunha (2019); Chen & Jin (2024), and other hybrid Feurer et al. (2015); Chen et al. (2021); Feurer et al. (2022) approaches. These systems addressed challenges like data sparsity and cold-start problems by leveraging user-item interaction patterns or metadata. While these methods have proven effective for traditional machine learning pipelines, they face challenges in scalability and lack the necessary mechanisms to address the complexity of LLM-based AI pipelines.

As more and more applications of LLM-based systems are being realized, the community has tried to solve this problem using the power of strong LLMs to automate hyperparameter optimization Zhang et al. (2023a); Liu et al. (2023), to automate the ML pipeline constructionZhang et al. (2023b), etc. HuggingGPT Shen et al. (2024) uses a GPT model to decide to route a given prompt to an appropriate model from a pool of models available on HuggingFace. This line of work is called LLM routing.

LLM routing methods dynamically select the most appropriate LLM for a specific query or task, mapping prompts to suitable models from a predefined pool, often using techniques such as contrastive learning Chen et al. (2024) or clustering Srivatsa et al. (2024) in embedding spaces to optimize routing decisions. The RouteLLM Ong et al. (2024) framework pioneered preference-driven routing, leveraging human feedback and data augmentation to train router models. However, its routing is limited to two models: a strong (mostly GPT-based) and a weak (mostly public) model, which is not suitable for a user who is unclear about selecting the right model from a large pool of public models available. In training-based routers Shnitzer et al. (2023); Lu et al. (2024), a model is learned from training data (e.g. past evaluations) to route the test prompts to the most appropriate model. The major drawback of this kind of model is that they need extensive training data, which is very expensive to obtain, and it is hard to generalize this training on new models. Recent advances include training free routing Guha et al. (2024); Zhao et al. (2024). Though these methods do not need extensive training data, they are not scalable to many LLMs, and the studies are conducted using a handful of models. Many of these works rely on LLMs' past performance on benchmarks,

but none of them propose any solution to minimize such evaluations to reduce the cost and carbon footprint.

Our proposed framework RELM, addresses these pain points by a) recommending LLMs from fastly growing public LLM repositories such as Huggingface. b) co-optimizing the LLM recommendation and evaluation problem by guiding the evaluation methods to minimize their cost of evaluation, and learning from it to improve the recommendations. While existing routing optimizes within small pools (10-20 models), RELM solves the upstream problem: selecting the optimal pool from 180K+ LLMs. RELM matches user-defined use cases against a *large* collection of Foundation Models using metadata-based similarity and partial benchmark performance data, filtering thousands of candidates down to a manageable subset (e.g., 10–20 models). Once the coarse-level filtering is complete, an LLM router (such as Guha et al. (2024)) can be applied to precisely choose the optimal LLM(s) for each incoming prompt, exploiting prompt-level signals and real-time performance metrics.

3 Problem Formulation

Our goal is to develop a system that can recommend the most suitable LLMs and relevant benchmarks while minimizing evaluation costs. This system must balance the exploration of new, potentially superior models with the exploitation of well-understood existing ones, all while continuously refining its recommendations based on new evaluation results.

Model and benchmark Recommendation Let Q and Q_v represent the input user query and corresponding vectorial representation, respectively. Let $M = [M_1, M2, \ldots, M_m]$ be the see of all available LLMs and $B = [B_1, B_2, \ldots, B_b]$ be the set of all possible benchmarks from different platforms such as HELM Liang et al. (2023) and LM Evaluation Harness Gao et al. (2024). We denote the metadata characteristics vector of model i by $C[M_i]$. Let $P_{ij} = P(M_i|B_j)$ denotes the performance of model i on benchmark. Let $\sigma_{model}(Q_v, C[M_i])$ and $\sigma_{benchmark}(Q, B_j)$ signifies the association of the query vector Q_v and model M_i.

Our objective is to: (a) maximize the relevance of recommended models and benchmarks to the user query; (b) identify models requiring further evaluation to estimate their expected performance; (c) recommend models that maximize expected performance on recommended benchmarks.

Model evaluation and need for co-optimization In order to approximate the expected performance of recommended models in the real world, the models need to be evaluated on the recommended benchmarks. In an already running system, it is likely that the performance is known for several <model, benchmark > pairs. For the remaining pairs, we need to compute $P_{ij} = P(M_i|B_j)$ in an efficient way.

The recommender system and evaluation platform need to guide each other in such a way, it improves the quality of recommendations and can run evaluations in an efficient way. A balance is essential between recommending models for evaluation (an expensive process) and recommending based on existing evaluations (which could exclude better-performing models).

Mathematically, this objective can be formulated as:

$$S^* = \underset{S \subseteq M}{\text{argmax}} \left\{ \beta \left(\sum_{i \in S} \left((\alpha)\sigma_{\text{model}}\left(Q_v, C[M_i]\right) + (1-\alpha) \sum_{j \in B} \sigma_{\text{benchmark}}\left(Q, B_j\right) \cdot P_{ij} \right) \right) - (1-\beta)|S| \right\} \tag{1}$$

where

- S^*: Optimal set of recommended models
- S: Subset of size $|S|$ comprising models selected from M for recommendation.
- $\beta \in [0, 1]$: Hyperparameter controlling the exploration (low β) and exploitation (high β) trade-off between considering more models and minimizing the cost of evaluation.
- $\alpha \in [0, 1]$: Weight balancing between semantic similarity and benchmark performance.

The objective function aims to identify the set of models S that maximizes the first quantity (term corresponding to β) while minimizing the size of S (corresponding to $1 - \beta$). More models are

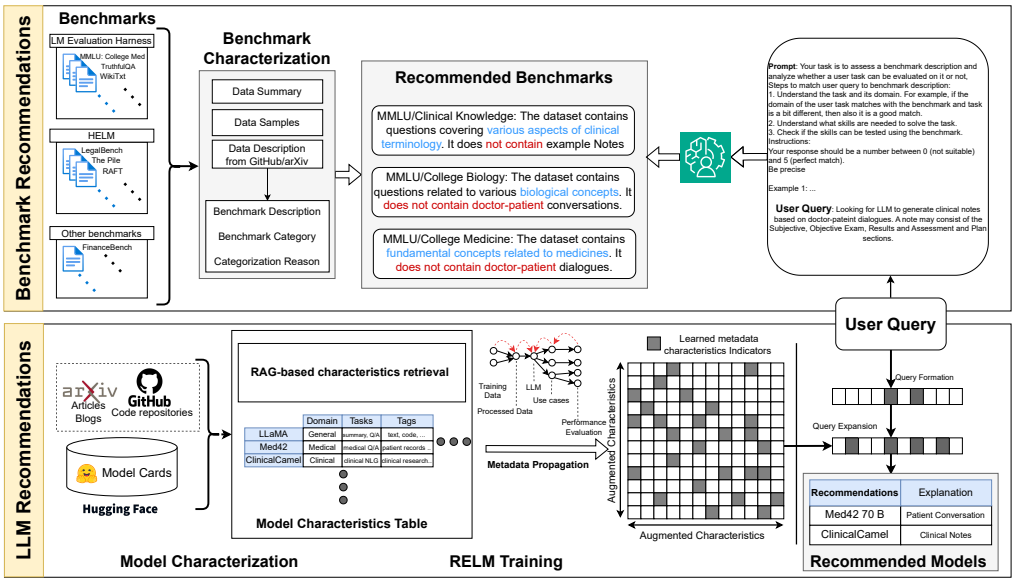

Figure 1: **RELM architecture** for recommending use-case specific benchmarks and LLMs. Both models and benchmarks are first characterized and then LLMs are recommended based on the metadata characteristics. Whereas the benchmarks are recommended based on the descriptive metadata.

recommended for large values of β, which require more evaluations. This setting will reduce the chances of missing the right LLM but incur more evaluation costs. On the other hand, lower β values make the setup more conservative and recommend fewer models with high confidence. Inside the first term of the objective function, α controls the importance of semantic similarity of the use case to model metadata characteristics vs. the model's performance on recommended benchmarks.

It can be noticed that solving the objective function requires exploring all possible subsets of models along with all benchmarks at once making this problem computationally expensive. In the solution section, we will discuss the proposed algorithm that makes use of a co-optimization framework of model recommendation and evaluation to solve this problem.

4 Proposed Solution: RELM, HERD and their Co-Optimization

We introduce RELM (Recommender Engine for Large Models) and HERD (Holistic Evaluation, Ranking, and Deciphering), a co-optimizing framework for recommending optimal LLMs while minimizing evaluation costs. This section details the components and their synergistic interaction.

4.1 LLM and Benchmark Recommendation

RELM and HERD act as two interdependent components in a continuous feedback loop. RELM drives the prioritization of evaluations in HERD, and HERD ensures that RELM's recommendations are always based on the latest and most relevant evaluations. This symbiotic relationship ensures that both systems evolve together. As shown in Figure 1, our proposed approach is as follows:

4.1.1 Model and Benchmark Characterization

Our framework leverages LLM metadata characteristics for generating recommendations. We realized that the metadata characteristics from HuggingFace model cards often lack important characteristics such as application domain and task details. To overcome this issue, we retrieve LLM characteristics from not just modelcards but also from associated blogs, research articles, and GitHub repositories. As shown in Figure 1, we employ a RAG-based pipeline with 20 questions covering domains, tasks, applications, and ethical considerations (complete list in Appendix) and extract required information from all the model-related resources. For benchmarks, we have rich information available in their content, so we take a different characterization approach. We begin by generating a benchmark description using OpenAI's GPT-4o model and append it with existing descriptions from

sources like GitHub or arXiv papers. Followed by selecting representative samples by clustering and random selection from each cluster.

This process results in LLMs characterized by one-hot vector representations and benchmarks described by detailed descriptions and representative samples.

4.1.2 RELM TRAINING

The RELM training process consists of two key steps:
Metadata Propagation: We observed that the original LLM metadata characteristics lack important information and can be enriched by bringing their complete context from their lineages. These lineage graphs consist of artifacts such as training data, processing steps, models, fine-tuning data, fine-tuned models, etc., as nodes that are connected by edges as per their relations in the workflow. Each artifact in the lineage graph can have its own characteristics. For example, training data can have its name and a readme file, including a summary of the data. Similarly, a learned model can have a list of tags, domains where it has been applied, a list of benchmarks it has been tested on, etc. We begin by propagating metadata characteristics back and forth across LLM lineage graphs. This process enriches each LLM's profile with expanded characteristics denoted by $C[M]$ derived from related artifacts in its lineage, including Training data, Fine-tuning data, base model information, etc.
Association Discovery: Following metadata propagation, we employ an indicator-based approach to uncover interesting associations between characteristics Kumar et al. (2023). This method identifies pairs of metadata characteristics that surprisingly co-occur more frequently than expected by chance. These discovered associations serve as hidden signals, which are later utilized to expand and enrich user queries during the recommendation phase. For example, a specific model characteristic such as "trained on tabular data" associated with the domain "molecular biology" indicates that the model is well-suited to understand gene expression data, even if this is not explicitly specified in the model properties.

This two-step process enables RELM to capture both explicit and implicit relationships within the LLM ecosystem, enhancing its ability to make nuanced and context-aware recommendations.

4.1.3 GENERATING RECOMMENDATIONS

The recommendation process begins with a user's input use case description, Q. We obtain its vectorial representation Q_v using SenBERT Reimers & Gurevych (2019) and identify the best matching characteristics from the set of all model characteristics. We use cosine similarity to compute the similarity between the user query and metadata characteristics. This set is then expanded using the learned indicators from the previous stage, resulting in $\sigma(Q_v, C[M_i])$, the association strength. Models with the highest association strength are potential recommendations.

For benchmark recommendations, we utilize a strong LLM (e.g., OpenAI's GPT-4o model) to analyze the use case and benchmark descriptions to capture characteristics such as task specificity, required skills, and domain relevance. Our prompt structure (detailed in the Appendix) evaluates these characteristics to identify the most relevant benchmarks. Following Equation 1, we compute $\sigma_{model}(Q_v, C[M_i])$ and $\sigma_{benchmark}(Q, B_j)$ for shortlisted models and benchmarks, and identify the entries in P_{ij} that need to be filled. In Equation 1, the number of recommendations is controlled by β. We fix it to maximum ten models based on the available resources.

4.2 CO-OPTIMIZATION OF LLM RECOMMENDATIONS AND EVALUATIONS

The LLMs recommended in the previous stage with missing entries in P_{ij} matrix require evaluation on the suggested benchmarks.

4.2.1 HERD PERFORMS MULTISTAGE LLM EVALUATION

Inspired by real-world scenarios where a large pool of options (e.g., job candidates) undergoes multiple filtering stages before final assessment, HERD uses a multistage pipeline to filter out less suitable models early in the process. This approach focuses computational resources on the most promising candidates as follows:

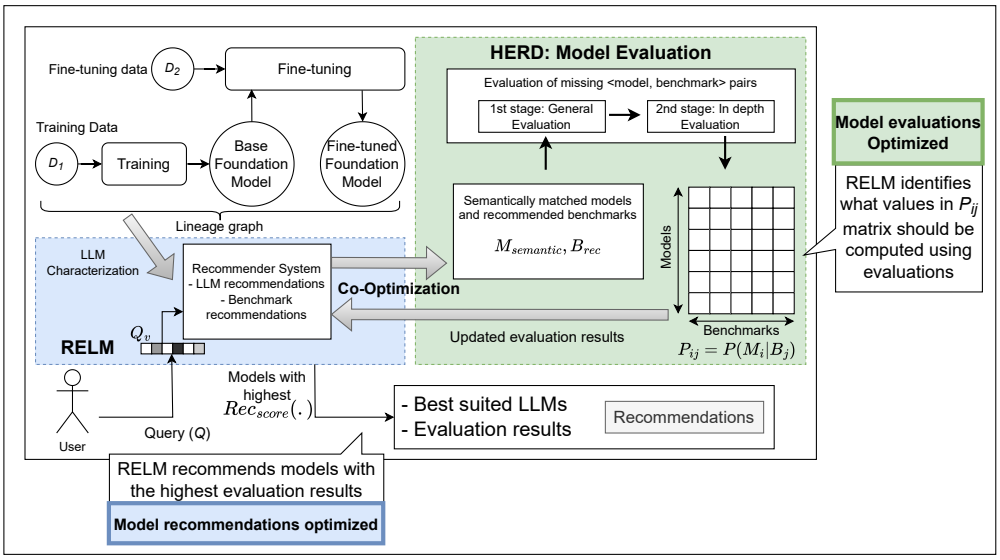

Figure 2: **Co-optimization of LLM recommendation and evaluation**: RELM guides the Model evaluation platforms with the right (model, benchmark) pairs to save their compute cost. These evaluations are used by RELM in order to improve the recommendations.

Stage One - Initial Fitness Assessment: HERD first examines the recommended models on suggested benchmarks to assess their general fitness. For instance, in a medical note generation use case, this stage might analyze models' performance on benchmarks like MMLU/Clinical_Knowledge and MMLU/College_Biology, as shown in Appendix section A.2.
Stage Two - Simulate Real Task: Models that pass the initial assessment undergo a more comprehensive, multi-dimensional evaluation using either real dataset or synthetic dataset that is close to the user queries in real task. Continuing with the medical example, this stage employs a LLM-as-a-judge, whose prompt is listed in Appendix section B, to assess domain-specific parameters such as medical accuracy and consistency.

4.2.2 Feedback Loop and Joint Optimization

The co-optimization process between RELM and HERD creates a powerful feedback loop that continuously improves both recommendation quality and evaluation efficiency:
Iterative P_{ij} Matrix Updates: Each HERD execution fills missing values in P_{ij}, which is crucial for guiding recommendations (as shown in Equation Equation 1). This ongoing update process ensures that RELM's recommendations become increasingly informed and accurate over time.
Guided Evaluation Priority: As RELM generates more recommendations, it provides HERD with prioritized (model, benchmark) pairs for evaluation. This guidance helps focus evaluation efforts on the most relevant and potentially impactful combinations. Thus, we optimize computational resource usage, allowing for broader coverage of the model space without exhaustive testing.
Dynamic Recommendation Refinement: The continual updates to P_{ij} allow RELM to refine its recommendations in real-time, adapting to new evaluation results and changing performance landscapes. The joint optimization process enables the system to learn from its recommendations and evaluations, continuously improving its understanding of model capabilities, benchmark relevance, and the relationships between them.

This symbiotic relationship between RELM and HERD creates a self-improving system that becomes more accurate and efficient over time, addressing the core challenges of LLM selection and evaluation in a rapidly evolving landscape. An additional advantage of co-optimizing the recommendation and evaluation processes is the mitigation of false positives that may arise during the recommendation phase. HERD conducts a comprehensive analysis of a model's suitability for the user-specified task, enabling it to flag models that may have been included in the recommendation list due to noise or misleading factors, such as ambiguous characteristic names.

Table 1: Description of the use cases from different domains

Domain	Use case Description	Task type
Healthcare	Recommend LLM to generate clinical notes with Subjective, Objective, Assessment and Plan sections, based on doctor-patient dialogues.	Open end text generation
Chemistry	Recommend LLM to perform property prediction task on molecules represented using SMILE strings	Classification
Finance	Recommend LLM to assess loan capability of a client based on their financial records.	Classification

5 EXPERIMENTS

To evaluate the effectiveness of RELM, HERD, and our co-optimization framework, we conducted a series of experiments across diverse domains. For experimentation purpose, we worked with a subset of top 10K downloaded models with the "text-generation" tag. This step ensures that we don't train our system on the models that are not relevant for most of the use cases. Note that all experiments in this section were conducted before October 2024 and newer models released after that date were not included. However, RELM is designed to update its database on a daily basis, ensuring that its real-world deployment can recommend the most recent models.

We designed three distinct experimental settings to assess the versatility and robustness of our approach, and compare the RELM results with (a) recommendations from HuggingFace search and (b) popular general-purpose LLMs. They are summarized in Table 1.

5.1 TEST USE CASES FROM VARYING DOMAINS

Open-ended text generation: Application in Healthcare we focused on the challenge of identifying an LLM capable of generating accurate SOAP (Subjective, Objective Exam, Assessment, and Plan) notes from doctor-patient dialogues. This task is challenging as it requires strong natural language understanding, medical terminology and clinical reasoning. The ideal LLM must be able to extract relevant medical information, understand the context of the patient's condition, and synthesize this information into a professionally formatted SOAP note. This requires balancing language skills and specialized medical knowledge, making the selection of the appropriate LLM crucial for ensuring the accuracy and usefulness of the generated notes in a clinical setting.

Classification of complex unstructured data: Applications in Chemistry our experiment centered on the task of identifying an LLM capable of classifying molecules as BACE inhibitors Guo et al. (2023) or non-inhibitors based on their SMILE string representations. This task is challenging due to its specialized nature, requiring an LLM that understands SMILE syntax and possesses deep molecular chemistry knowledge. The challenge lies in finding an LLM that can bridge the gap between string processing and specialized chemical knowledge, making the selection process critical for ensuring accurate and reliable predictions in drug discovery and molecular research contexts.

Classification of unstructured data: Applications in Finance we focused on the challenge of identifying an LLM capable of assessing financial records to determine loan eligibility. This task is challenging due to the complexity of financial data and the need for fair, accurate lending decisions. The ideal LLM must grasp financial concepts, regulatory requirements, and risk assessment while making balanced, unbiased judgments. The challenge lies in selecting an LLM that can process diverse financial data while ensuring ethical, compliant, and sound lending recommendations.

5.2 BASELINE METHODS

We employ two baseline methods to compare with RELM-recommended models for a given task:
Recommendations using HuggingFace search engine[2]: We utilize the Full-Text search service provided by HuggingFace, a widely used platform, to obtain their recommended models for each task and compare RELM recommended models with them.
Popular models: We identify a set of "popular models" by selecting those that appear on multiple leaderboards and rank highly in HuggingFace's default rankings.

[2]URL:https://huggingface.co/search/full-text?type=model

By comparing against these models, we aim to demonstrate that complex, domain-specific use cases often require specialized model selection beyond what general-purpose models can offer.

5.3 Extension of RELM to LLM routing

Recognizing the potential synergy between RELM's coarse-level model recommendation and fine-grained LLM routing methods, we build upon our open-text generation experiment in the Healthcare domain by implementing LLM routing Guha et al. (2024) on top of RELM-recommended models. By combining RELM's ability to narrow down the model search space with the precision of fine-grained routing techniques, this allows us to explore a hierarchical approach to model selection and query handling. The results of this extended experiment, presented in the following section.

6 Results and Analysis

Our experiments across three domains show that RELM effectively recommends optimal LLMs that consistently outperform baseline methods.

6.1 Three Use Case Studies

Table 2: Comparison of RELM-recommended models with HuggingFace recommendations for SOAP note generation in healthcare. Metrics represent HERD's second-stage evaluation results.

	Model	Average	Medical Accuracy	Relevance Pertinence	Medical Consistency	Ethical	Clarity
Popular Models	Vicuna-13B-v1.5	4.79	4.7	4.45	4.9	4.9	**5**
	Llama-2-70B-chat	4.61	4.3	4.4	4.45	5	4.9
	Mistral-7B-Instruct-v0.2	4.51	4.2	4.1	4.6	4.85	4.8
HuggingFace recommended	openbiollm-llama-3-70b	3.89	3.4	3.6	3.7	4.7	4.05
	openbiollm-llama-3-8b	1.42	1.1	1.1	1.1	2.7	1.10
	talktoaiZERO	1.28	0.9	0.9	1.0	2.7	0.90
RELM recommended Models	m42-health-70B	**4.95**	**5**	**4.95**	**5**	4.8	**5**
	ClinicalCamel-70B	4.83	4.75	4.65	4.75	**5**	**5**
	MedleyMD-7B	4.78	4.8	4.7	4.8	**5**	4.6

In the clinical note generation task, models recommended by RELM outperformed both popular models and HuggingFace's recommendations across HERD stage-2 evaluation metrics (see Table 2). The domain-specific LLMs suggested by RELM achieved above 4.8 (out of 5) average scores, signficantly higher than any other LLMs in the comparison. Our top recommendation, "m42-health-70B" Christophe et al. (2024), is an instruction-tuned version of the LLaMa2 model. It uses medical flashcards, exam questions, and open-domain dialogues, making it highly useful for our task.

Table 3: Accuracy comparison of RELM-recommended models, popular models, and HuggingFace recommendations for BACE inhibitor classification in chemistry.

	Model	Accuracy
Popular Models	Vicuna-13B-v1.5	0.60
	Llama-2-70B-chat	0.50
	Mistral-7B-Instruct-v0.2	0.42
	Llama-3.1-8B-Instruct	0.50
	Llama-3.1-70B	0.58
HuggingFace Recommended Models	Synthia-70B-v1.2b	0.54
	Tess-XS-Creative-v1.0	0.60[1]
	Mixtral-8x22B-Instruct-v0.1	0.54
RELM recommended Models	ChemDFM-13B-v1.0	**0.76**
	ChemLLM-20B-Chat-SFT	0.62

[1] The model returned response for only 20% of the input queries, and accuracy is computed only on the set where it returned a prediction.

Table 3 presents the accuracy comparison for the BACE inhibitor classification task in the chemistry domain. The results show that RELM-recommended models achieve higher accuracy compared to both popular models and HuggingFace recommendations. This underscores RELM's effectiveness in identifying models capable of handling specialized tasks requiring deep domain knowledge, such as interpreting SMILE strings and predicting molecular properties.

Table 4: Accuracy comparison of RELM-recommended models, popular models, and HuggingFace recommendations for loan assessment in finance.

	Model	Accuracy
Popular Models	gpt-4o-mini	0.36
	gpt-4o-2024-05-13	0.63
	Meta-Llama-3.1-405B-Instruct	0.60
	claude-3-5-sonnet-20240620	0.8
	Meta-Llama-3.1-8B-Instruct	0.33
HuggingFace Recommended Models	Phi-3.5-mini-instruct	0.35
	llawma-fact-enrich[1]	0.1
RELM Recommended Models	Mistral-7B-Mortgage-Loans	0.82
	process-loans	**0.84**
	DISC-FinLLM	0.81
	Mistral-7B-Banking-v2	0.81

[1] The model returned the answer by random guessing, and the explanation is consistent with all the test data.

Table 4 demonstrates RELM's capability to identify models that effectively process and assess financial data for accurate loan eligibility evaluations. Additionally, our stage-1 HERD evaluation (detailed in the Appendix, Table 7) offers valuable early insights. The benchmarks selected in stage-1 are strong predictors of final performance, efficiently filtering out models with weak domain-specific abilities. For instance, HuggingFace-recommended models like `Phi-3.5-mini-instruct` show relative low scores on FinanceBench (0.417) and MMLU Accounting (0.57), indicating a potential lack of necessary domain-specific strengths for accurate loan assessments. These early-stage findings assist RELM in maintaining a high recall of top-performing candidates.

The benchmark recommendation phase of the proposed framework offers a significant cost-saving advantage by reducing the need for exhaustive exploration. Even widely used benchmark suites, such as LM Eval Harness and HELM, require models to be evaluated on over 100 benchmarks, many of which may not be relevant to the user-specified use case. In contrast, our framework identifies the most relevant benchmarks, typically around three to five, allowing for a more efficient evaluation process while minimizing computational costs.

6.2 Integration with LLM Routing

Table 5: Performance comparison of RELM-powered LLM routing (using Smoothie Guha et al. (2024)) versus routing with HuggingFace Search recommended models for healthcare text generation."

Recommender Method	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore F1
HuggingFace Search + Smoothie	0.39	0.14	0.22	0.04	0.84
RELM + Smoothie	**0.52**	**0.26**	**0.34**	**0.11**	**0.87**

To demonstrate the potential of integrating RELM with fine-grained routing techniques, we extended our healthcare experiment to include LLM routing. Table 5 compares the performance of RELM-powered routing against routing using HuggingFace Search recommended models. The results show significant improvements across all metrics (ROUGE-1, ROUGE-2, ROUGE-L, BLEU, and BERTScore F1) when using RELM-recommended models for routing. This suggests that RELM not only identifies suitable models for a given task but also provides a strong foundation for more advanced query handling techniques.

7 CONCLUSION

In this paper, we introduced RELM (Recommender Engine for Large Models), HERD (Holistic Evaluation, Ranking, and Deciphering), and their joint optimization to address the critical challenge of selecting optimal Large Language Models (LLMs) for specific tasks. Our solution effectively navigates the vast landscape of publicly available LLMs, to identify suitable models across diverse domains such as healthcare, chemistry, and finance. Experimental results demonstrate that RELM-recommended models outperform both popular general-purpose models and those recommended by existing platforms, particularly in specialized tasks requiring deep domain knowledge.

Our work advocates for co-optimizing recommendation and evaluation processes, to enhance recommendation quality while controlling evaluation costs. The extension of RELM to LLM routing showed significant improvements, opening important future research directions. While we focused primarily on recommendation quality, future work could explore integrating cost optimization, especially considering recent LLM routing methods designed for this purpose. As the LLM landscape continues to expand, our framework offers a scalable solution for intelligent model selection, enabling users to harness the full potential of LLMs across a wide range of real-world applications.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Xiaoyu Chen and Ran Jin. Lori: Local low-rank response imputation for automatic configuration of contextualized artificial intelligence. *IEEE Transactions on Industrial Informatics*, 2024.

Yi-Wei Chen, Qingquan Song, and Xia Hu. Techniques for automated machine learning. *ACM SIGKDD Explorations Newsletter*, 22(2):35–50, 2021.

Clement Christophe, Praveenkumar Kanithi, Prateek Munjal, Tathagata Raha, Nasir Hayat, Ronnie Rajan, Ahmed Al Mahrooqi, Avani Gupta, Muhammad Umar Salman, Marco AF Pimentel, et al. Med42-evaluating fine-tuning strategies for medical llms: Full-parameter vs. parameter-efficient approaches. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*, 2024.

Tiago Daniel Sá Cunha. Recommending recommender systems: tackling the collaborative filtering algorithm selection problem. *Repositório Aberto da Universidade do Porto*, 2019.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. In *The Twelfth International Conference on Learning Representations*, 2024.

Alexey Dukhanov, Artem Smetatnin, and Anna Luthenko. A recommender system for machine learning pipelines' design to recognize objects in video frames as a learning tool for training data scientists. In *2024 International Russian Automation Conference (RusAutoCon)*, pp. 881–886. IEEE, 2024.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *Journal of Machine Learning Research*, 23 (261):1–61, 2022.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework

for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Ryan Green. Applying deep learning techniques to assist bioinformatics researchers in analysis pipeline composition. Master's thesis, University of Cincinnati, 2023.

Neel Guha, Mayee F Chen, Trevor Chow, Ishan S Khare, and Christopher Re. Smoothie: Label free language model routing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023.

Tarun Kumar, Arpit Shah, Ashish Mishra, Suparna Bhattacharya, Arun Mahendran, Ted Dunning, and Glyn Bowden. From roots to fruits: Exploring lineage for dataset recommendations. In *Proceedings of the Second ACM Data Economy Workshop*, pp. 41–47, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

Fei Liu, Xi Lin, Zhenkun Wang, Shunyu Yao, Xialiang Tong, Mingxuan Yuan, and Qingfu Zhang. Large language model for multi-objective evolutionary optimization. *arXiv preprint arXiv:2310.12541*, 2023.

Yixin Liu, Kejian Shi, Katherine He, Longtian Ye, Alexander Richard Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. On learning to summarize with large language models as references. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8639–8656, 2024.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1964–1974, 2024.

Mandana Mazaheri, Gregory Kiar, and Tristan Glatard. A recommender system for scientific datasets and analysis pipelines. In *2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)*, pp. 1–8. IEEE, 2021.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

Tal Shnitzer, Anthony Ou, Mirian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. In *Annual Conference on Neural Information Processing Systems*, 2023.

Kv Aditya Srivatsa, Kaushal Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from LLM routing. In Shabnam Tafreshi, Arjun Akula, João Sedoc, Aleksandr Drozd, Anna Rogers, and Anna Rumshisky (eds.), *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*, pp. 124–134, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.insights-1.15. URL https://aclanthology.org/2024.insights-1.15/.

Chengrun Yang, Yuji Akimoto, Dae Won Kim, and Madeleine Udell. Oboe: Collaborative filtering for automl model selection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1173–1183, 2019.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Michael R Zhang, Nishkrit Desai, Juhan Bae, Jonathan Lorraine, and Jimmy Ba. Using large language models for hyperparameter optimization. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023a.

Shujian Zhang, Chengyue Gong, Lemeng Wu, Xingchao Liu, and Mingyuan Zhou. Automl-gpt: Automatic machine learning with gpt. *arXiv preprint arXiv:2305.02499*, 2023b.

Zesen Zhao, Shuowei Jin, and Z Morley Mao. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*, 2024.

A Appendix

A.1 Model Characterization

The list of questions below is used to characterize models.

1. What are the tasks of the model? Consider tasks such as understanding text, answering questions, recognizing objects in images, etc.
2. What is the primary domain of applications of this model, such as general, healthcare, finance, legal, science, etc.?
3. Does this model perform better when prompts are augmented with examples?
4. Can this model effectively handle sensitive information?
5. What kind of data can the model understand? Text, images, structured data, etc.
6. What is the domain of training data? List few associated tags.
7. Is the model trained to follow user instructions?
8. What is the context length of the model?
9. What is the hardware configuration used to train the model?
10. What is the required hardware configuration needed to inference from this model?
11. What is the license category of this model?
12. Can the model do batch processing?
13. How does the model ensure fairness and avoid bias in its outcomes?
14. Does the model provide users with control over their data and privacy settings?
15. How well does the model manage data security, including privacy storage and transmission?
16. What processes are in place for auditing the AI model's compliance and ethical practices?

17. Can the model protect the data used for training?
18. Assess the level of transparency regarding the model's internal operations, especially how decisions are made, and data is processed.
19. What are the measures for built-in safety features and ethical guardrails in the model?
20. Is this model fine-tuned using a base model? If yes, name the base model.

A.2 HERD STAGE-1 RESULTS

A.2.1 MEDICAL USE CASE

Table 6: Stage-1 HERD results for the Medical Use Case. The table reports scores on five benchmarks: Clinical knowledge, College biology, College medicine, Professional medicine, and Anatomy.

Group	Model	Clinical knowledge	College biology	College medicine	Professional medicine	Anatomy
Popular Models	vicuna-13b-v1.5	0.637	0.611	0.578	0.540	0.481
	Llama-2-70b-chat	0.637	0.750	0.601	0.577	0.518
	Mistral-7B-Instruct-v0.2	0.671	0.694	0.589	0.617	0.570
	mpt-30b-instruct	0.516	0.534	0.450	0.448	0.422
	Llama-2-7B-chat	0.535	0.520	0.398	0.455	0.429
HuggingFace Recommended Models	OpenBioLLM-70B	0.929	0.938	0.857	0.938	0.839
	OpenBioLLM-8B	0.761	0.842	0.680	0.782	0.698
	talktoaiZERO	0.789	0.806	0.682	0.787	0.682
RELM Recommended Models	med42-70B	0.743	0.840	0.688	0.798	0.674
	ClinicalCamel-70B	0.698	0.792	0.670	0.713	0.622
	MedleyMD-7b	0.720	0.729	0.658	0.683	0.607

In this section, we present the stage-1 HERD evaluation results for the medical use case. In this initial evaluation phase, each model was assessed on five distinct benchmarks capturing various aspects of medical competence: *Clinical knowledge*, *College biology*, *College medicine*, *Professional medicine*, and *Anatomy*. These early results serve as a preliminary checkpoint to guide further, more detailed evaluations in HERD stage-2 (see Table 2).

Table 6 reports the stage-1 scores for the benchmarks selected by RELM. Notably, RELM-recommended models such as `med42-70B`, `ClinicalCamel-70B`, and `MedleyMD-7b` exhibit competitive performance across the benchmarks, which is consistent with the superior stage-2 performance observed in Table 2. This grouping reinforces the effectiveness of our metadata-driven selection strategy in identifying models well-suited for the medical use case.

A.2.2 FINANCE USE CASE

Table 7: Stage-1 HERD results for the Finance Use Case on selected benchmarks

Group	Model	MMLU Account	MMLU Econometrics	MATH	FinanceBench	LegalBench (Finance Split)
Popular Models	GPT-4o-mini	0.63	0.61	0.802	0.551	0.631
	GPT-4o-05-13	0.73	0.74	0.829	0.761	0.728
	Claude-Sonnet-3.5	0.78	0.77	0.813	0.791	0.701
	Llama3.1-405B-Instruct	0.74	0.73	0.827	0.759	0.708
	Llama3.1-70B-Instruct	0.69	0.69	0.783	0.639	0.721
	Llama3.1-8B-Instruct	0.61	0.62	0.703	0.537	0.618
HuggingFace Recommended Models	Phi-3.5-mini-instruct	0.57	0.56	0.681	0.417	0.564
	llawma-fact-enrich	0.40	0.29	0.193	0.501	0.591
RELM Recommended Models	Mistral-8B-Mortgage-Loan	0.68	0.71	0.608	0.698	0.638
	DISC-FinLLM-13B	0.66	0.70	0.620	0.719	0.610

Table 7 presents the stage-1 HERD results for the finance use case (loan assessment) across five benchmarks selected by RELM. The stage-1 evaluation provides several useful insights that support our co-optimization strategy. For example, the HuggingFace-recommended model `llawma-fact-enrich` shows very low performance in quantitative reasoning (MATH: 0.19) and economic understanding (Econometrics: 0.29). This justifies its exclusion from stage-2 evaluation and is consistent with its low final task performance (approximately 0.1 accuracy in Table 4). Similarly, while `Phi-3.5-mini-instruct` achieves a moderate MATH score (0.681), its lower

scores on FinanceBench (0.417) and MMLU Accounting (0.57) suggest that it may lack the necessary domain-specific strengths for effective loan assessment.

Among the Popular Models, balanced performance is observed across benchmarks, with Claude-Sonnet-3.5 obtaining the highest FinanceBench score (0.791) and GPT-4o-05-13 leading in MATH (0.829). In contrast, the RELM Recommended Models exhibit more targeted strengths. For instance, Mistral-8B-Mortgage-Loan shows a strong FinanceBench score (0.698) relative to its overall parameter count, and DISC-FinLLM-13B achieves competitive performance on the LegalBench (Finance Split) benchmark (0.610) compared to larger popular models. These focused strengths contribute to their superior stage-2 performance, where they achieve task accuracies in the range of 0.82–0.84, as shown in Table 4.

Overall, these results highlight HERD's effectiveness in early-stage filtering by identifying models with domain-specific capabilities. Notably, the FinanceBench metric appears to be a more reliable predictor of final loan assessment accuracy than the MATH score, as illustrated by the differences in performance between models such as Mistral-8B-Mortgage-Loan and Llama3.1-70B-Instruct. By reducing the number of candidates advanced to stage-2, this early-stage discrimination helps lower evaluation costs while maintaining a high recall of top performers. The co-optimization framework, therefore, enhances both the quality and efficiency of model selection by focusing evaluations on candidates that demonstrate domain-adapted strengths.

B EXPERT PROMPT IN HERD STAGE 2

HERD stage 2 prompt structure

System Prompt: You are a very experienced doctor reviewing a clinical note based on a dialogue between a doctor and a patient. The note was generated by a large language model (LLM).

Instructions:

```
    The diaglogue is provided below:
<start_of_dialogue>
{dialogue}
<end_of_dialogue>

The generated clinical note by a large language model (LLM) is
as follows:
<start_of_note>
{response}
<end_of_note>

Your task is to review the generated note and rate the note in
multiple dimensions.

Please provide your review in the following format
(IMPORTANT: DO NOTE USE BOLD in any part of your review):
1. Accuracy of Medical Information
 - Does the note accurately represent the dialogue between the doctor
 and the patient?
 - Are the medical facts and information in the note correct?
 - Are there any discrepancies or errors in the medical details
 provided?
Rating: 1 to 5 (1: Very Inaccurate, 5: Very Accurate)
2. Relevance and Pertinence
 - Is the information provided relevant to the patient's condition and
 the dialogue?
 - Are all the key points from the dialogue captured in the note?
 - Is there any irrelevant or unnecessary information included?
Rating: 1 (Very Irrelevant) to 5 (Very Relevant)
3. Medical Consistency and Cohesiveness
 - Is the information medically coherent and logically structured?
 - Does the note maintain consistency in the patient's symptoms,
 diagnosis, and treatment plan?
 - Are there any contradictions or inconsistencies in the patient's
 medical history or treatment recommendations?
Rating: 1 (Very Inconsistent) to 5 (Very Consistent)
4. Ethical Considerations
 - Is patient confidentiality and privacy maintained?
 - Are there any ethical concerns regarding the content or
 presentation of the information?
 - Does the note respect cultural, social, and personal sensitivities?
Rating: 1 (Very Unethical) to 5 (Very Ethical)
5. Clarity and Comprehensibility
 - Is the note clearly written and easy to understand?
 - Is the note structured in a way that is easy to follow?
Rating: 1 (Very Unclear) to 5 (Very Clear)
6. Additional Comments
Please provide any additional observations or suggestions for
improvement:
```