ScaleDoc: Scaling LLM-based Predicates over Large Document Collections

HENGRUI ZHANG*, Tsinghua University, China
YULONG HUI*, Tsinghua University, China
YIHAO LIU, Tsinghua University, China
HUANCHEN ZHANG†, Tsinghua University, China

Predicates are foundational components in data analysis systems. However, modern workloads increasingly involve unstructured documents, which demands semantic understanding, beyond traditional value-based predicates. Given enormous documents and ad hoc queries, while Large Language Models (LLMs) demonstrate powerful zero-shot capabilities, their high inference cost leads to unacceptable overhead. Therefore, we introduce ScaleDoc, a novel system that addresses this by decoupling predicate execution into an offline representation phase and an optimized online phase. In the offline phase, ScaleDoc leverages a LLM to generate semantic representations for each document. Online, for each query, it adaptively trains a lightweight proxy model on these representations to filter the majority of documents, forwarding only the ambiguous cases to the LLM for final decision. Furthermore, ScaleDoc proposes two core innovations to achieve significant efficiency: (1) a contrastive-learning-based framework that trains the proxy model to generate reliable predicating decision scores; (2) an adaptive cascade mechanism that determines the effective filtering policy while meeting specific accuracy targets. Our evaluations across three datasets demonstrate that ScaleDoc achieves over a 2× end-to-end speedup and reduces expensive LLM invocations by up to 85%, making large-scale semantic analysis practical and efficient.

CCS Concepts: • **Information systems → Data management systems**; **Document representation**.

Additional Key Words and Phrases: Large Language Models, Semantic Predicate, Document Collections

ACM Reference Format:
Hengrui Zhang, Yulong Hui, Yihao Liu, and Huanchen Zhang. 2026. ScaleDoc: Scaling LLM-based Predicates over Large Document Collections. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 229 (June 2026), 26 pages. https://doi.org/10.1145/3802106

1 Introduction

Predicates are essential for selecting relevant data in relational databases, search engines, and big-data systems. Traditionally, these systems have excelled at value-based predicates (e.g., CITY = 'New York'). However, modern analytical tasks increasingly require querying enormous corpora of unstructured documents based on their semantic meaning [16, 33]. For example, medical researchers might look for all publications that "developed novel psychotropic medications", or enterprises

*Both authors contributed equally to this research.
†Huanchen Zhang is also affiliated with the Shanghai Qi Zhi Institute. Corresponding author.

Authors' Contact Information: Hengrui Zhang, Tsinghua University, Beijing, China, zhanghen22@mails.tsinghua.edu.cn; Yulong Hui, Tsinghua University, Beijing, China, huiyl22@mails.tsinghua.edu.cn; Yihao Liu, Tsinghua University, Beijing, China, liuyihao24@mails.tsinghua.edu.cn; Huanchen Zhang, Tsinghua University, Beijing, China, huanchen@tsinghua.edu.cn.

ACM 2836-6573/2026/6-ART229
https://doi.org/10.1145/3802106

may need to analyze reports where customers are "expressing dissatisfaction with service quality." Such queries require a deep contextual understanding that goes beyond simple keyword matching.

Handling these ad hoc semantic queries in large-scale analytics presents a significant challenge. Conventional machine learning (ML) models are not scalable for this purpose due to the extensive effort in engineering and data labeling required for each new task. While Large Language Models (LLMs) offer a zero-shot solution with their remarkable general capabilities, their high computational cost creates a major barrier. The expense of running LLM inference on millions of documents for every query makes this approach impractical for widespread adoption [2, 15].

Real-world semantic judgments are often ambiguous, making perfect accuracy costly and impractical. Consequently, many systems prioritize scalable efficiency by aiming for a specific accuracy target [18, 29, 38, 47]. To achieve this trade-off, some systems, such as NoScope [18] and PPs [29], use lightweight proxy models to filter easier cases. However, these proxies are designed for traditional small models and specific tasks. They lack zero-shot flexibility and fail to bridge the vast capability gap with massive LLMs. More recent LLM-centric solutions, such as FrugalGPT [4] and LOTUS [34], use smaller LLMs (e.g., GPT-3.5 or LLaMA-8B) as filtering proxies. While more flexible, these smaller LLMs remain too computationally expensive for truly large-scale applications involving millions of documents.

A critical inefficiency in LLM-based approaches is the repetitive re-processing of entire documents for every ad hoc query, incurring substantial and redundant computational costs. A key insight is to shift these expensive, document-centric LLM computations to a one-time offline phase to reduce the burden during online processing. We, therefore, propose ScaleDoc, a system designed for efficient LLM-based predicate execution over enormous document corpora. ScaleDoc decouples the execution into two phases. The one-time *offline representation* phase creates a rich semantic representation of each document using an LLM. Then, when an ad-hoc query arrives, the *online processing* phase trains a lightweight, query-specific proxy model that uses these representations to rapidly filter the documents. The model identifies and forwards only the most ambiguous documents to the powerful but expensive LLM, achieving significant efficiency gains while maintaining accuracy.

While this architecture is promising, basic engineering integration is insufficient. Basic approaches often fail to address the fundamental capacity mismatch, inherent in scaling LLMs for complex scenarios. Therefore, the effectiveness of ScaleDoc depends on overcoming two key challenges. First, the lightweight proxy must be trained to provide reliable **decision scores**. Lacking the oracle LLM's deep semantic understanding, standard training methods often yield ambiguous scores that fail to distinguish positive from negative cases. Under such proxy ambiguity, most documents would still be forwarded to the expensive oracle. To address this, we propose a new design principle: **query-aware distribution shaping**. Guided by this principle, we developed a framework based on contrastive learning that trains the proxy to capture fine-grained semantics and produce well-behaved predicating scores.

The second challenge is that the system must determine an effective **filtering criterion** for each ad hoc query. Because the relationship between indeed accuracy and decision scores is unknown for a new query, it is difficult to establish a filtering threshold that meets a user's accuracy target while minimizing cost. We tackled this with an **adaptive cascade mechanism** that performs online calibration to shape query-specific score distributions, and then uses an optimized algorithm to determine the ideal filtering thresholds.

We evaluated ScaleDoc across three diverse datasets, where it significantly outperforms existing baselines. On average, ScaleDoc achieves over 2× end-to-end performance speedup and reduces costly LLM invocations by up to 85%.

Our paper makes the following contributions:

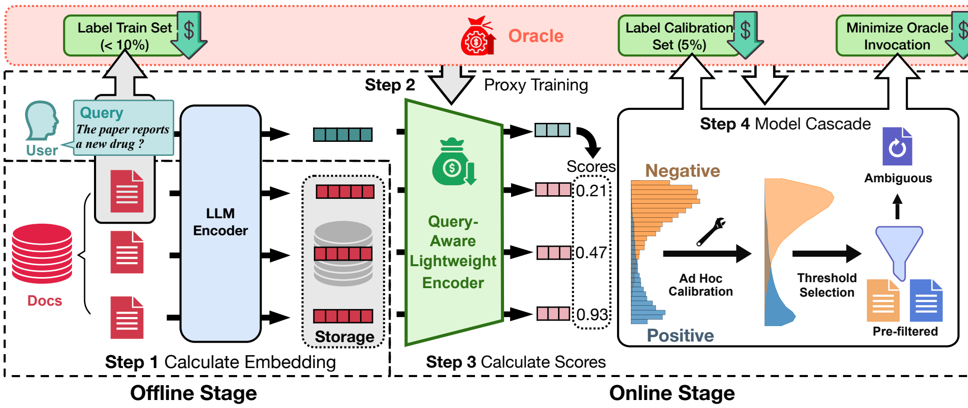


Fig. 1. A detailed workflow of ScaleDoc – ScaleDoc adapts pre-calculated semantic embeddings for query-specific online processing. The online process comprises a query-aware lightweight encoder and a subsequent cascade workflow.

- We propose ScaleDoc, a novel system that decouples LLM execution into a one-time offline representation phase and an optimized online query phase.
- We introduce a contrastive learning strategy to train a lightweight, query-aware proxy model that provides reliable predicating decision scores.
- We design an adaptive online calibration mechanism with an optimized filtering algorithm to guarantee user-specified accuracy while minimizing calls to the expensive oracle LLM.

2 System Overview

ScaleDoc provides an efficient solution for executing semantic predicates over large-scale documents. This section defines the target workloads, introduces the system's architecture, and outlines the core challenges to overcome.

2.1 Workload Specification

A semantic predicating workload consists of a large document collection D, and a set of queries Q. Each query, $q \in Q$, is defined by two components: a natural language predicate and a user-specified accuracy target. Formally, we use an SQL-like syntax to express these queries. For instance, a query to find all medical papers in PubMed that introduce a new drug would be:

```
SELECT * FROM PubMed
WHERE "The paper introduces a new drug"
WITH accuracy_target = 0.90
```

For each such query, the final task is to evaluate the predicate against every document $d \in D$ and assign a binary label (positive or negative), while meeting the specified accuracy target.

2.2 System Architecture

ScaleDoc adopts an offline-online architecture, enabling efficient execution by decoupling query processing stages. An overview of the ScaleDoc pipeline is illustrated in Figure 1.

The offline stage is a one-time, compute-intensive process. For each document, we use a small-scale LLM (e.g. with 7B parameters) to generate a semantic embedding, which is then stored for online use. This approach offers two main benefits. First, it leverages the expressive power of the LLM to produce semantically rich document representations. This provides a high-quality foundation, allowing lightweight processing in subsequent online stages. Second, it front-loads the necessary yet expensive computations of LLM. By pre-computing and storing these embeddings, ScaleDoc can efficiently reuse them across countless ad hoc queries, eliminating the need to repeatedly run the LLM.

The online stage employs a proxy-cascade architecture to handle ad hoc queries efficiently. Upon receiving a new query, ScaleDoc will train a query-specific lightweight proxy model. This model then rapidly evaluates the documents, assigning each a decision score, indicating the likelihood of the positive predicate. To train this model, we need to first sample a small fraction (e.g. 5%) of the documents, and obtain ground-truth predicating labels by calling a powerful oracle LLM (e.g. GPT-4o [32]). After the proxy model, a subsequent cascade filter uses the proxy decision scores to decide the high-confidence and low-confidence documents. The proxy's judgment is adopted for the high-confidence set, whereas the low-confidence set is forwarded to the oracle LLM for a final judgment. This hierarchical approach strategically minimizes oracle invocations, ensuring scalable and cost-effective query execution.

2.3 Core Challenges

While this offline-online and proxy-oracle architecture is promising, its effectiveness hinges on overcoming two critical challenges:

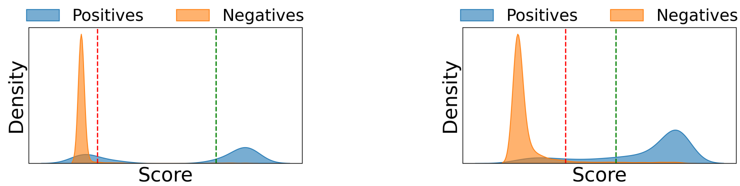
Reliable Proxy Model. The efficiency of the cascade filter depends on the quality of the decision scores produced by the lightweight proxy model. First, a naive model might fail to capture the fine-grained semantics between the query and documents, leading to unreliable proxy predictions. Second, a poorly trained model may generate ambiguous scores that do not clearly distinguish positive and negative documents. Such an ambiguity would force a large fraction of instances to be categorized as "uncertain", leading to excessive invocations of the expensive oracle LLM and diminishing the system's performance. Therefore, the first challenge is to develop a well-behaved proxy model, capable of generating accurate and decisive scores, enabling maximum data reduction.

Ad Hoc Cascade. Given the proxy model's scores, the cascade filter must determine the decision thresholds to separate documents into high-confidence (to be filtered) and low-confidence (to be verified by the oracle) sets. The goal is to meet the user-specified accuracy target while minimizing the number of oracle calls. However, in an ad hoc setting, neither the query-specific data distribution nor the ground-truth labels are known beforehand. This lack of prior knowledge makes it non-trivial to select effective thresholds online. Therefore, the second challenge is to design an efficient online calibration mechanism that can dynamically determine the effective thresholds, ensuring the final accuracy target is met with minimal oracle overhead.

To address these two challenges, ScaleDoc introduces a novel contrastive-learning-based approach to train a robust proxy model (Section 3) and proposes an efficient calibration workflow for the model cascade (Section 4). Together, these contributions ensure both high data reduction and robust predicating accuracy.

3 Query-Aware Model Training

The online efficiency of ScaleDoc hinges on a lightweight, query-aware proxy model. For each ad hoc query, this model is trained to emulate the semantic judgments of the LLM oracle. It takes precomputed document representations as input and outputs a decision score for each document, indicating the likelihood of satisfying the semantic predicate. This scoring enables a cascaded filtering process: high-confidence positive and negative documents are filtered, while only low-confidence, ambiguous documents are sent to the LLM oracle. Therefore, these decision scores, used to reduce oracle invocations, are paramount to overall system performance.



(a) Llama-3.2-3B logprobs

(b) ScaleDoc

Fig. 2. Example score distributions of different proxies, with low and high data reduction rate.

However, training an effective lightweight proxy model is non-trivial. Basic approaches, such as the naive binary classifiers in prior work [29, 47], struggle to capture query-specific semantics. Our preliminary experiments with a more advanced query-fused regression MLP also yielded unreliable scores.

To address this, we argue that the training process must be explicitly designed to produce scores with a well-behaved distribution. We achieve this through a contrastive learning framework. In Section 3.1, we detail the desirable properties of this score distribution. Section 3.2 then describes how ScaleDoc leverages our contrastive learning mechanism to generate the scores.

3.1 Desirable Distribution on Decision Scores

The primary goal of the lightweight model is data reduction, filtering more documents based on their decision scores. For clarity, we assume higher scores indicate a higher confidence in being positive and lower scores indicate being negative. Therefore, the low-score and high-score thresholds determine the filtering.

In practice, a high data reduction rate is not always guaranteed. Figure 2a illustrates how a poorly structured score distribution (*logprobs* of Llama-3.2-3B [43]'s first response token) can severely hinder data reduction. Specifically, the real positive class exhibits two peaks, appearing in both high and low scores. The attempt to set a filter threshold in the low-score region (e.g., the left red line) would **misclassify** many positives, violating the accuracy target and forcing a conservative and ineffective threshold. In contrast, Figure 2b illustrates a desirable distribution with a clear separation. Only minimal positive items fall below the red line threshold, allowing effective filtering of the true negatives.

To consistently achieve effective filtering, we propose three essential properties for the decision scores:

1) Smoothness. The distribution of decision scores should be continuous and smooth, without abruptions or discontinuities. This ensures the threshold-selection algorithms can operate stably and consistently. We have observed that this fundamental property fails with some naive proxies such as Kernel Density Estimation (KDE).

2) Semantic monotonicity. A document that is more semantically satisfying the query should be assigned a higher decision score, and vice versa. This prevents reward-hacking of deterministic score ranges, ensuring a meaningful and reliable ranking.

3) Bipolarity. The score distribution must be strongly polarized, with positive and negative documents clustering distinctly at the high and low ends of the score spectrum. This clear separation is the primary driver of high data reduction.

Achieving these desirable properties is challenging. While LLM-based embeddings offer rich semantics, they are static and not optimized for the specific semantic distinctions of an ad-hoc query. Directly training a lightweight model (e.g., MLP) on these embeddings typically yields

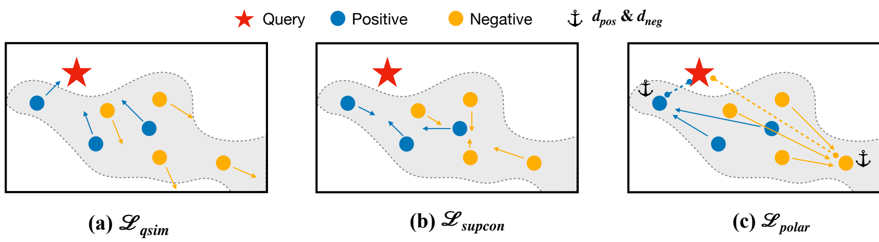


Fig. 3. Illustration of the objectives adopted in training ScaleDoc's Query-Aware Encoder.

inconsistent score distributions. To overcome this and derive discriminative query-specific patterns, we employ a contrastive learning-based approach, which excels at capturing fine-grained, task-specific distinctions.

3.2 A Contrastive Learning-based Approach

ScaleDoc employs a contrastive-learning-based framework, designed to refine pre-computed embeddings into tailored and semantic-aligned decision scores. The core is a **lightweight encoder** $E(\cdot)$, which utilizes the MLP structure, mapping both the document and query into a shared **latent space**. Following this, the final decision score is the cosine similarity between the encoded representations of the query and documents. This score is inherently smooth, and through our tailored training, optimized to be semantically monotonic and bipolar.

Formally, given a query q and a document d, we first obtain their high-dimensional semantic embeddings from an LLM encoder, denoted as $\mathbf{e}_q \in \mathbb{R}^D$ and $\mathbf{e}_d \in \mathbb{R}^D$. ScaleDoc then employs a lightweight encoder, $E(\cdot) : \mathbb{R}^D \to \mathbb{R}^l$, mapping these embeddings into a latent space $\mathbb{R}^l$:

$$\mathbf{z}_q = E(\mathbf{e}_q), \quad \mathbf{z}_d = E(\mathbf{e}_d)$$

The decision score, $s(q, d)$, for the query-document pair is defined as the cosine similarity between their latent representations:

$$s(q, d) = \text{sim}(\mathbf{z}_q, \mathbf{z}_d) = \frac{\mathbf{z}_q \cdot \mathbf{z}_d}{||\mathbf{z}_q|| ||\mathbf{z}_d||}$$

For this task T, consisting of a query q and a collection of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, the complete set of decision scores can be expressed as:

$$S(T) = S(q, \mathcal{D}) = \{s(q, d_i) \mid d_i \in \mathcal{D}\}$$

To effectively shape the latent space and meet the required properties, ScaleDoc trains the encoder in two distinct phases. This two-phase strategy is crucial because jointly optimizing all properties simultaneously can lead to conflicting training signals and results in observed performance decline. Phase 1 first establishes the correct semantic relationship as the foundation. Then Phase 2 refines the spectrum of the embedding space to create a clear separation between positive and negative cases. There are three objectives during the overall training process, depicted in Figure 3.

Training Phase 1: Semantic Monotonicity This phase aims to build the foundational semantic relationship between the documents and the query. We use a contrastive loss $\mathcal{L}_{qsim}$, inspired by dense passage retrieval [20]. In our training, the query embedding $\mathbf{z}_q$ acts as an anchor.

The objective is to pull positive document embeddings (d^+) closer to the anchor while pushing negative ones (d^-) away in the latent space, as shown in Figure 3(a).

Formally, given a query q, each input mini-batch $\mathcal{D}'$ comprises m positive documents $\{d_i^+\}_{i=1}^m$ and $n-m$ negative documents $\{d_j^-\}_{j=1}^{n-m}$, hence $\mathcal{D}' = \{d_i^+\} \bigcup \{d_j^-\}$. The training loss is defined as:

$$\mathcal{L}_{qsim}(q, \mathcal{D}') = -log \frac{\sum\limits_{i=1}^{m} e^{\text{sim}(q, d_i^+)/\tau}}{\sum\limits_{d \in \mathcal{D}'} e^{\text{sim}(q,d)/\tau}} \tag{1}$$

Here, τ is a temperature hyperparameter. The objective utilizes negative log-likelihood to maximize similarity between queries and positive documents, while penalizing negatives. This relocates positives closer to the query within the latent space.

A crucial distinction from previous work [20] is that we train our encoder **dynamically** for each new query task. This allows the model to learn query-specific semantics rather than a single static matching function. It is essential in our setting: our online proxy model is intentionally quite lightweight for efficiency, lacking the general capacity of large models. By specializing the proxy for each query, we enable it to perform effectively despite its limited size.

However, this training phase alone is insufficient to guarantee the robust bipolarity. Without an explicit separation margin, we observe that local neighbors (mixed positive and negative cases) often relocate together. This results in distribution overlap and class collapse, which hinders effective data filtering.

Training Phase 2: Enforcing Bipolarity To address the limitations of Phase 1, the second phase explicitly shapes the latent space to form a bipolar distribution. Using the same encoder and training data, we introduce two loss functions, $\mathcal{L}_{supcon}$ and $\mathcal{L}_{polar}$.

First, $\mathcal{L}_{supcon}$ adopts supervised-contrastive-learning to encourage intra-class clustering [22], pulling documents of the same label together, as shown in Figure 3(b). The loss is formulated as:

$$\mathcal{L}_{supcon}(\mathcal{D}') = -\sum_{i=1}^{n} \frac{1}{|U(i)|} \log \frac{\sum_{d_p \in U(i)} e^{\text{sim}(d_i, d_p)/\tau}}{\sum_{d_k \in A(i)} e^{\text{sim}(d_i, d_k)/\tau}} \tag{2}$$

Within a mini-batch, for each anchor document d_i , $U(i)$ denotes documents sharing the same label with d_i, and $A(i)$ comprises all other documents . With this objective, we can create a discriminative embedding space that minimizes intra-class variance, crucial for the proxy to learn a clear decision boundary.

Second, $\mathcal{L}_{polar}$ introduces a novel mechanism to explicitly enforce a bipolar manifold and enlarge the margin between classes. The intuition of $\mathcal{L}_{polar}$ is selecting a *bellwether* sample for positives and negatives in each mini-batch to guide the direction of clustering. A bellwether is distinguished by the closest (positive) or furthest (negative) distances relative to the query inside each mini-batch. For a given mini-batch and query, we define the bellwethers as:

$$d_{pos} = \underset{d_i \in \{d^+\}}{\text{argmin}} \ \text{sim}(q, d_i), \quad d_{neg} = \underset{d_j \in \{d^-\}}{\text{argmax}} \ \text{sim}(q, d_j)$$

$\mathcal{L}_{polar}$ then uses these bellwethers as anchors, pulling positive documents towards d_{pos} and and negative documents towards d_{neg}.

$$\mathcal{L}_{polar}(q, \mathcal{D}') = -log \frac{\sum\limits_{i=1}^{m} e^{sim(d_{pos}, d_i^+)/\tau}}{\sum\limits_{d \in \mathcal{D}'} e^{sim(d_{pos}, d)/\tau}} - log \frac{\sum\limits_{j=1}^{n-m} e^{sim(d_{neg}, d_j^-)/\tau}}{\sum\limits_{d \in \mathcal{D}'} e^{sim(d_{neg}, d)/\tau}} \tag{3}$$

Bellwethers can be different across mini-batches, but $\mathcal{L}_{polar}$ ensures a consistent relocation. The progressive process explicitly enlarges the separation margin between the positive and negative poles. This constructs a highly bipolar manifold, enhancing the efficacy of threshold-based filtering, as shown in Figure 3(c).

Collectively, these two training phases and three loss functions effectively shape the latent space to produce decision scores satisfying our proposed properties. For training data, we heuristically sample a small subset of documents (e.g. 5%) and use the oracle LLM to generate ground-truth labels (more implementation details in Section 5).

4 Model Cascade

After the proxy procedure, ScaleDoc uses a powerful but expensive LLM oracle to resolve uncertain documents through the cascade component. Model cascade aims to meet the user-specified accuracy while minimizing oracle invocations by filtering most documents and only forwarding the ambiguous cases to the oracle.

In ad hoc settings, the lack of prior knowledge about query-specific labels and data distribution poses a significant challenge, often leading to suboptimal data reduction in existing methods [19]. ScaleDoc addresses this with a novel calibration workflow and an optimized filtering algorithm. Leveraging well-behaved decision scores from our lightweight model (Section 3), ScaleDoc achieves superior data reduction while robustly achieving accuracy targets.

This section proceeds as follows. Section 4.1 formulates the cascade task as a constrained optimization problem. Section 4.2 introduces our ad hoc calibration method, which enables accurate performance estimation from a small sampled subset. Section 4.3 details the algorithm for selecting optimal filtering thresholds. Section 4.4 presents the theoretical discussion of the accuracy maintenance achieved by our algorithms with bounded error.

4.1 Problem Formulation

For a given query task T, the lightweight proxy model produces decision scores for all documents, denoted by $S(T)$. Each score is a cosine similarity within the interval $[0, 1]$, where the higher one indicates stronger semantic agreement with the query. To filter out high-confidence instances, the cascade component should select two thresholds, a lower bound l and an upper bound r. The filtering logic is as follows:

- Documents with scores in $(r, 1]$ are classified as **positive**.
- Documents with scores in $[0, l)$ are classified as **negative**.
- Documents with scores in $[l, r]$ are deemed **ambiguous** and are sent to the LLM oracle for final decision.

The primary goal is to minimize the fraction of documents sent to the LLM oracle, while satisfying the user-specified accuracy target α. We define this fraction as the unfiltered rate u, controlled by the lower bound and upper bound:

$$u(l, r) = \frac{|\{s_i \in S(T) \mid l \le s_i \le r\}|}{|S(T)|} \tag{4}$$

The final accuracy, $\mathrm{Acc}(l, r)$, is determined by the correctness of the lightweight model in the filtered regions, combined with the perfect accuracy of the oracle on the unfiltered region $[l, r]$.

This leads to the following optimization problem, where $\mathcal{K}$ represents the search space for (l, r), with $0 \le l < r \le 1$.

$$\min_{(l,r)\in\mathcal{K}} u(l, r) \quad \text{s.t. } \mathrm{Acc}(l, r) \ge \alpha \tag{5}$$

Algorithm 1 Ad hoc Calibration Workflow

Given: Decision scores $S(T)$, Oracle LLM
$bins \leftarrow \text{Discretize}(S(T))$ *// Discretize score range into bins for stratified sampling and analysis*
$S'(T) \leftarrow \text{StratifiedSample}(S(T))$ *// Sample representatively from each bin*
$S'_P \leftarrow \{s_i \mid s_i \in S'(T) \land \text{Oracle}(q, d_i) = \text{positive}\}$
$S'_N \leftarrow \{s_i \mid s_i \in S'(T) \land \text{Oracle}(q, d_i) = \text{negative}\}$
$PDF_P \leftarrow \text{Smooth}(\text{DE}(\text{Jitter}(S'_P)))$ *// Reconstruct score distribution for positive class*
$PDF_N \leftarrow \text{Smooth}(\text{DE}(\text{Jitter}(S'_N)))$ *// Reconstruct score distribution for negative class*
return $PDF_P, PDF_N, bins$

Solving this problem in an ad hoc setting is challenging, because the function $\text{Acc}(l, r)$ is unknown without full ground truth statistics. Therefore, a calibration process is required to estimate the accuracy, followed by the thresholds selection algorithm.

4.2 Ad-hoc Calibration

To solve the optimization problem in (5), we must obtain the accuracy $\text{Acc}(l, r)$ for any given threshold pair (l, r). However, in the ad hoc setting, the ground-truth labels for the full collection are not available. Therefore, ScaleDoc relies on a small oracle-labeled sample to estimate the accuracy and calibrate the cascade.

However, this poses a critical challenge: small samples may fail to capture the true global relationship between labels and proxy scores. The sampled score distributions can deviate substantially from the true global distributions, which may lead to poorly calibrated threshold (l, r) and fail to meet the accuracy target. This issue stems from two key limitations: (1) simple random sampling might under-represent documents in low-density regions and cause information loss [19]; (2) sample randomness may introduce stochastic noise. While a sufficiently large sample could mitigate the estimation bias, the high cost of oracle labeling makes this infeasible.

To address this, ScaleDoc introduces a robust calibration workflow (Algorithm 1), which reconstructs the global score distributions from a small sample. It comprises two main stages: **stratified sampling** and **distribution reconstruction**. Our experiments demonstrate that with a modest sample size of 5%, ScaleDoc can achieve great cascade performance.

Stratified Calibration Sampling. As stated above, uniform random sampling may fail to capture low-density score regions and introduce noise, especially with smaller sample sizes. To address these issues, we propose to employ stratified sampling. We discretize the entire score range into a series of bins. Then, we sample from each bin in proportion to its population in the global set $S(T)$. This ensures that the sampled subset $S'(T)$ preserves the relative density of the global distribution, preventing low-density regions from being entirely omitted.

Distribution Reconstruction. After obtaining the labeled sample $S'(T)$, we reconstruct continuous and robust score distributions for positive and negative instances. The core objective is to faithfully mirror the global distributions $S(T)$, optimized through the following mathematical process:

1) Jittering for Information Recovery. To counteract information loss, especially in low-density regions, we first apply random jittering. Stratified sampling with a modest rate can cause bins with few samples to become empty. Ignoring these gaps encourages overconfidence in these score ranges, creating a risk factor for subsequent threshold selection. Jittering addresses this by introducing low-density, random data into these empty bins, ensuring that this information is not lost before subsequent modeling.

2) Density Estimation (DE) via Linear Interpolation. Next, we construct the continuous Probability Density Functions (PDF_P and PDF_N) from the discrete jittered samples. We employ *linear interpolation* for this density estimation task, providing virtually distortion-free modeling. This process yields a continuous probabilistic interface that facilitates subsequent accuracy calculations via density integration rather than discrete sample counting. This allows querying over arbitrary scores for an estimated density value. Compared to methods like Kernel Density Estimation (KDE), linear interpolation provides a more faithful representation of the empirical distribution, particularly in low-density regions.

3) Smoothing. Finally, to mitigate anomalous noise and information loss from sampling randomness, we apply the Moving Average (MA) approach. MA is a simple soft computing method that smooths the distribution by applying mean pooling within a fixed-size sliding window, which allows for a better capture of the general distributional features. Applying an MA filter reduces spikes and breakpoints, providing an additional layer of smoothing to reveal the underlying global trend more effectively.

With the above optimizations, the reconstructed distributions serve as a continuous and robust foundation for the subsequent threshold selection process. The final outputs of Algorithm 1 are the distributions of positives and negatives in the form of PDFs, and the discretization of the score range *bins*.

4.3 Threshold Selection

After calibrating the score distributions, we determine the optimal filtering threshold (l, r), as defined in (5). The objective is to minimize the unfiltered rate $u(l, r)$ while satisfying the user-specified accuracy α. A basic brute-force search over all possible threshold pairs brings quadratic complexity. Therefore, we introduce an efficient algorithm that leverages the monotonic properties of the calibrated score distributions. The core insight is that any optimal solution must lie on the Pareto-frontier of the feasible set defined by the accuracy constraint, $\mathrm{Acc}(l, r) \geq \alpha$. Points off this frontier are suboptimal as their unfiltered rate can be improved by tightening the thresholds.

Our approach, outlined in Algorithm 2, efficiently identifies the optimal thresholds by tracing the frontier. The potential thresholds, denoted as *steps*, are discrete bin boundaries in our calibration. The algorithm proceeds in three main phases:

1) Boundary Identification: Ensuring the accuracy target, the algorithm first determines the two extreme points of the feasible frontier: (l_0, r_s) and (l_s, r_0). Here, l_0 is the tightest (highest) lower bound possible when the upper bound is most conservative (r_s), and r_0 is the lowest upper bound when the lower bound is most conservative (l_s). This step effectively bounds the search space to a one-dimensional path.

2) Frontier Traversal: Starting from (l_0, r_s), the algorithm iteratively constructs a path P that approximates the feasible frontier. It greedily moves towards the other extreme point (l_s, r_0) by tightening either l or r at each step, ensuring the accuracy constraint α is never violated.

3) Optimal Point Selection: Since the unfiltered rate $u(l, r)$ is monotonically decreasing with respect to the size of the filtered region, the optimal point that minimizes u must lie on the path P. The algorithm simply evaluates $u(l, r)$ for all points in P and returns the pair (l_t, r_t) that yields the minimum value.

This algorithm reduces complexity from quadratic to linear with respect to the number of discretization steps.

4.4 Theoretical Guarantee for Accuracy

The cascade workflow selects optimal thresholds (l, r) based on a small sampled subset S' (with sampling ratio p) to satisfy a target accuracy α. These thresholds are then applied to filter the

Algorithm 2 Thresholds Selection

Given Reconstructed CDF_P, CDF_N; discretization *bins*
$steps \leftarrow bins.edges$ *// Initialize thresholds search space*
$l_s, r_s \leftarrow steps.\text{first}, steps.\text{last}$
function SELECTTHRESHOLDS (α)
  $l_0, r_0 \leftarrow l_s, r_s$ *// Initialize thresholds*
  // 1. Find the tightest l_0 with $r = r_s$
  for l **from** *steps.first* **to** *steps.last* **do**
    if $\text{Acc}(l, r_s) \geq \alpha$ **then** $l_0 \leftarrow l$ **else break**
  end for
  // 2. Find the tightest r_0 with $l = l_s$
  for r **from** *steps.last* **to** *steps.first* **do**
    if $\text{Acc}(l_s, r) \geq \alpha$ **then** $r_0 \leftarrow r$ **else break**
  end for
  // 3. Construct frontier path P from (l_0, r_s) to (l_s, r_0)
  $l, r \leftarrow l_0, r_s$
  $P \leftarrow \{(l_0, r_s)\}$
  while $(l, r) \neq (l_s, r_0)$
    $l_{next}, r_{next} \leftarrow l + bins.size, r - bins.size$
    if $\text{Acc}(l_{next}, r) \geq \alpha$ **then**
      $l \leftarrow l_{next}$
    else
      $r \leftarrow r_{next}$
    $P \leftarrow P \bigcup \{(l, r)\}$
  // 4. Find the best threshold pair on the path P
  $l_t, r_t \leftarrow \underset{(l,r)\in P}{\text{argmin}}\ \text{UNFILTERED}(l, r)$
  return l_t, r_t

entire document collection S. In this section, we provide a theoretical analysis to prove that this sample-based cascade could maintain the end-to-end accuracy on the global population.

Let the full dataset S have size N, with document decision scores distributed over $[0, 1]$. To guarantee that the true global accuracy Acc_S exceeds α with a high confidence level $(1 - \delta)$, we must derive a safety margin ϵ for our estimates on subset S'. We denote $F_S^{\pm}(\cdot)$ and $F_{S'}^{\pm}(\cdot)$ as the cumulative mass functions for the global population and the sample set, respectively, where $F_S^{\pm}$ and $F_{S'}^{\pm}$ represent their total masses.

Formulating the Accuracy Constraint. Under our cascade logic, documents with scores in $[l, r]$ are routed to the oracle. Therefore, proxy errors only occur in the filtered regions: unverified false negatives have a mass of $F^+(l)$ (true positives scored $< l$), and unverified false positives have a mass of $F^- - F^-(r)$ (true negatives scored $> r$). The global F1-score can thus be written as:

$$\text{Acc}(l, r) = \frac{2(F^+ - F^+(l))}{2(F^+ - F^+(l)) + (F^- - F^-(r)) + F^+(l)} \geq \alpha \tag{6}$$

To facilitate bounding the error, we algebraically reformulate this fractional constraint into a linear one. Let $\mathcal{T}(l,r)$ represent the weighted sum of the erroneous tail masses:

$$\mathcal{T}(l,r) = \left(1 - \frac{\alpha}{2}\right) F^+(l) + \frac{\alpha}{2}\left(F^- - F^-(r)\right) \tag{7}$$

The accuracy target $\text{Acc}(l,r) \geq \alpha$ is mathematically equivalent to bounding this tail mass: $\mathcal{T}(l,r) \leq (1-\alpha)F^+$.

Proposition 1. *Let $|S'| = pN$ be the sample size. For any candidate thresholds (l,r) where $0 \leq l < r \leq 1$, and $\delta > 0$, there exists a safety margin $\epsilon > 0$ such that if the sample satisfies:*

$$\mathcal{T}_{S'}(l,r) \leq (1-\alpha)F^+_{S'} - \epsilon$$

then the true global accuracy satisfies:

$$P[Acc_S(l,r) \geq \alpha] \geq 1 - \delta$$

Proof. Let $\text{Oracle}(i) = \mathbb{1}[i\text{-th document is positive}]$ and s_i be the score of the i-th document. For any pair (l,r), the empirical functional $\mathcal{T}_{S'}(l,r)$ corresponds to the sample mean of i.i.d. random variables Z_i over the pN samples, defined as:

$$Z_i = \left(1 - \frac{\alpha}{2}\right) \mathbb{1}[\text{Oracle}(i) = 1 \wedge s_i < l] + \frac{\alpha}{2} \mathbb{1}[\text{Oracle}(i) = 0 \wedge s_i > r] \tag{8}$$

We apply Bernstein's inequality to bound the estimation error between the sample mean $\mathcal{T}_{S'}(l,r)$ and the true global mean $\mathcal{T}_S(l,r)$:

$$P\left[|\mathcal{T}_{S'}(l,r) - \mathcal{T}_S(l,r)| \geq \epsilon_1\right] \geq 1 - \delta_1 \tag{9}$$

where the deviation bound is $\epsilon_1 = \sqrt{\frac{4\sigma_Z^2 \ln(2/\delta_1)}{pN}} + \frac{4(1-\alpha/2)\ln(2/\delta_1)}{3pN}$, and $\sigma_Z^2 = \text{Var}(Z_i)$ denotes the variance of Z_i,

Similarly, for the right-hand side of the constraint (7), we bound the estimation error of the positive class mass F^+:

$$P\left[|(1-\alpha)F^+_{S'} - (1-\alpha)F^+_S| \geq \epsilon_2\right] \geq 1 - \delta_2 \tag{10}$$

where $\epsilon_2 = (1-\alpha)\left(\sqrt{\frac{4\sigma_P^2 \ln(2/\delta_2)}{pN}} + \frac{4\ln(2/\delta_2)}{3pN}\right)$, and σ_P^2 is the variance of the sampled positives.

By applying a union bound to combine Eq. (9) and (10), and setting $\delta_1 = \delta_2 = \delta/2$, we establish that if the observed sample constraint holds $\mathcal{T}_{S'}(l,r) \leq (1-\alpha)F^+_{S'} - \epsilon$, then global constraint $\mathcal{T}_S(l,r) \leq (1-\alpha)F^+_S$ holds with probability at least $1-\delta$. This implies $P[\text{Acc}_S(l,r) \geq \alpha] \geq 1 - \delta$.

The global safety margin $\epsilon = \epsilon_1 + \epsilon_2$ evaluates to:

$$\epsilon = \left(\sqrt{\sigma_Z^2} + (1-\alpha)\sqrt{\sigma_P^2}\right)\sqrt{\frac{4\ln(4/\delta)}{pN}} + \frac{(8-6\alpha)\ln(4/\delta)}{3pN} \tag{11}$$

Discussion. Proposition 1 demonstrates that the global error is bounded by the variances of proxy scores, theoretically justifying our contrastive learning approach (in Section 3.1), which optimizes for low-variance distributions. Furthermore, the discretization in Algorithm 2 acts as a conservative buffer to cover the margin ϵ. Collectively, these provide a theoretical guarantee that ScaleDoc can generalize from the proxy sample to the full dataset, maintaining the end-to-end accuracy with high confidence.

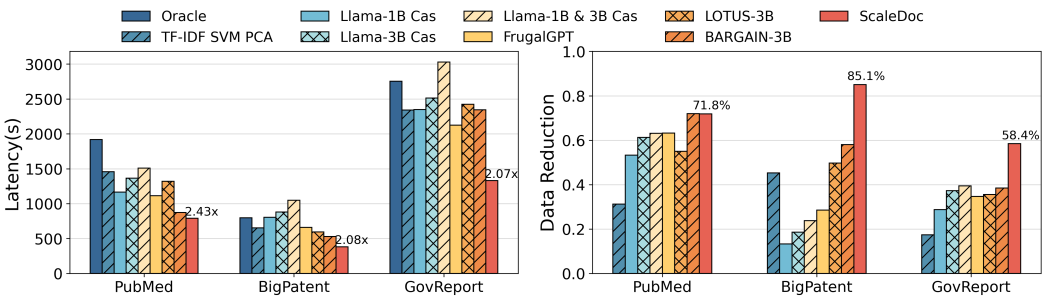


Fig. 4. End-to-end latencies and data reduction rate – We evaluate ScaleDoc and other baselines with accuracy target $\alpha = 0.90$. The data reduction rate measures the percentage of data that does not require the LLM oracle call, indicating the cost-saving.

5 Implementation Details

In this section, we discusses several key design decisions and optimizations within the system.

Training Set Optimization. In the online phase, the initial step of training is to sample a small subset of documents for oracle labeling. However, randomly sampling may get quite imbalanced classes, resulting in a biased and ineffective proxy. To address this, we implement a fallback-style rebalancing method. If the initial sample is quite skewed, we augment the data by adding Gaussian noise to the existing minority embeddings and assigning them the corresponding label, thus creating a more balanced training set.

Model Training. The proxy encoder is a 3-layer perceptron (MLP), mapping the embeddings into a latent space. Following standard practice in contrastive learning [5, 10], we append a projector head $Proj(\cdot)$ after the encoder, during the training. This projector head is discarded during inference. The training mini-batch consists of multiple document embeddings and the single query embedding, forces the encoder to optimize the shared latent space for both.

In Section 3.2, training-phase-2 optimizes a joint loss: $\mathcal{L}_2 = \lambda \cdot \mathcal{L}_{supcon} + (1 - \lambda) \cdot \mathcal{L}_{polar}$, where the hyperparameter λ balances the two losses. Across all experiments, we empirically set $\lambda = 0.2$. We found the final performance is not sensitive to this parameter, and extensive tuning did not yield significant improvements.

Model Cascade. The calibration step relies on discretization granularity, which determines the number of bins to partition the proxy score ranges (the *steps* in Algorithm 2). We empirically set this granularity to 64, balancing distribution accuracy with sample representativeness per bin.

6 Evaluation

6.1 Experimental Setup

Workloads. To evaluate performance across diverse settings, we use three real-world document collections: PubMed [7], BigPatent [41], and GovReport [13]. Given the lack of benchmarks for collection-level semantic predicates, we manually crafted 20 distinct natural language queries for each collection, covering a wide range of semantic characteristics and data selectivity. Each collection contains 10,000 documents, and the ground truth labels for all query-document predicating pairs are generated by GPT-4o [32]. Table 1 provides a summary of the datasets and illustrative queries. We utilize **F1 score as the accuracy metric** throughout our paper, which is more robust than naive accuracy for handling imbalanced data.

Table 1. Dataset Characteristics

Dataset	Content	Avg. Word Count	Example Predicating Query
PubMed	Abstracts from medical papers	413	Report efficacy of a certain medicine?
BigPatent	Summaries on U.S. patent documents	129	Introduce inventions of cross-sectional technology?
GovReport	Reports from government	621	About environment protection?

Baselines. We compare ScaleDoc against several representative baselines, which accelerate ML-powered predicates in analytical data systems. The oracle model for all systems, which also serves as the ground truth labeler, is **GPT-4o** [32].

1) Oracle Only. This baseline avoids any filtering and processes every document directly using the oracle LLM. We prompt the LLM to output the binary class labels to indicate the predicate decision.

2) Probabilistic Predicates (PPs) [29, 47]. This approach uses traditional lightweight machine learning models as proxies. For each new problem, the PPs are trained to output binary classification results with confidence interface. It includes the following component choices:

- Text Representation: Bag-of-Words (BoW) and TF-IDF, both are standard text encoding schemes.
- Dimensionality Reduction: Principal Component Analysis (PCA) and Feature Hashing (FH).
- Classifier: Support Vector Machines (SVM) and Kernel Density Estimation (KDE).

These models are implemented using scikit-learn [35]. We explore various combinations of them and report the best performing choices.

3) LLM Cascade [4, 34]. This category employs smaller, cost-effective LLMs as proxies to filter easy instances, selectively deferring only uncertain cases to the oracle. In our setup, Llama-3.2-1B or 3B models serve as proxies, while GPT-4o acts as the oracle. We utilize the vLLM engine [23] to ensure high-throughput proxy inference. We evaluate four distinct cascade strategies:

- **Basic Cascade.** We implement standard cascade pipelines, including single-proxy configurations (denoted as *1B-Cas* and *3B-Cas*) and a multi-hop chain (*1B⟶3B⟶Oracle*). These baselines utilize the log-probability of the generated token as the confidence score. To determine the filtering thresholds, they adopt the same cascade strategy described in Section 4.
- **FrugalGPT** [4]. This approach employs a scoring model (DistilBERT [39]) to predict the reliability of the proxy LLM's response. It then uses a constrained optimization formulation to maximize accuracy within a given budget. We integrate both 1B and 3B models as proxies, and FrugalGPT would autonomously build a cascading pipeline. To ensure a fair comparison with our accuracy-target setting, we profile the cost-accuracy curve of FrugalGPT and report the minimum oracle usage required to achieve the target accuracy.
- **LOTUS** [34]. With a database perspective, LOTUS implements semantic operators and adopts a variant of the SUPG algorithm [19] to accelerate predicates. Similar to ScaleDoc, it relies on an independent sampling procedure to estimate statistics and derive cascade thresholds over the sampled subset.
- **BARGAIN** [48]. This recent framework provides theoretical statistical guarantees for cascade flows. It introduces different optimization targets (i.e., precision, recall and accuracy target). To align with our setting, we compare against its accuracy-target (AT) strategy, which minimizes

Table 2. Estimated average computational cost (FLOPs) per query - The FLOPs are normalized to 10,000 documents. Each column present the overall averaged invocation frequency of the model. "Total" represents the sum of all components (the labeling cost is omitted).

Models(→) FLOPs	Our Proxy 2T	1B 10P	3B 27P	Oracle >500P	Total
ScaleDoc	**1**×	-	-	**0.28**×	**140P**
1B & 3B Cas	-	1×	0.42×	0.59×	316P
LOTUS-3B	-	-	1×	0.61×	332P
BARGAIN-3B	-	-	1×	0.44×	249P
Oracle	-	-	-	1×	500P

oracle usage while satisfying a correctness constraint. Since its AT strategy optimizes for exact-match accuracy (ScaleDoc use F1 score by default), we implement an exact-match variant of ScaleDoc to align the accuracy metrics and report the normalized results.

4) Direct Embedding Matching. For additional ablation, we adopt a design from embedding-based retrieval systems [12]. This method computes a similarity score between the query and each document using an *off-the-shelf* embedding model. This score serves directly as the proxy value for cascade filtering (see results in Section 6.4).

Experimental Details. Each workload operates on a collection of 10,000 documents. For methods requiring proxy fine-tuning (i.e., ScaleDoc and PPs), we sample 1,000 documents (10%) for training and calibration. The remaining 9,000 documents are then processed online. Baselines that do not require training (i.e., oracle-only and LLM-cascade) just process the entire collection online. For the offline representation phase, ScaleDoc uses NvEmbed [24], derived from Mistral-7B [17], to pre-compute the document embeddings. Our online lightweight proxy model is implemented as a 3-layer MLP. We run LLM inference on a single NVIDIA A10 GPU, while the oracle LLM (GPT-4o) is accessed via the Azure-OpenAI API, subject to a rate limit of 150k tokens per minute.

6.2 End-to-End Performance

We evaluate end-to-end online performance of ScaleDoc against the baselines. Figure 4 presents the average online latency and data reduction across three datasets, with a user-specified accuracy target of $\alpha = 0.90$ (α for all experiments will be 0.90 if not specified). The results show that ScaleDoc consistently and significantly outperforms all other approaches, achieving an average online **speedup of over 2**×. ScaleDoc also reduces the invocation of oracle LLMs up to **85%**, which means 6.6× cost-saving. Specifically, ScaleDoc achieves better oracle reduction in 44 out of 60 queries against BARGAIN-3B and outperforms LOTUS-3B in 53 out of 60 queries. The performance gain is driven by effective data filtering and the computational efficiency of our lightweight design.

Computational Efficiency. For a more holistic evaluation of ScaleDoc's cost-effectiveness, beyond latency and data reduction ratios, we analyze the total computation consumption. We estimate the Floating Point Operations (FLOPs) required per query, with the averaged document length as input (i.e., approximately 400 words). Table 2 details the invocation frequency of different models and the resulting total FLOPs (normalized to 10,000 documents). The results demonstrate that ScaleDoc achieves the lowest total computational cost (140P), significantly outperforming other strategies. This efficiency stems from the synergy of two factors: the negligible training and

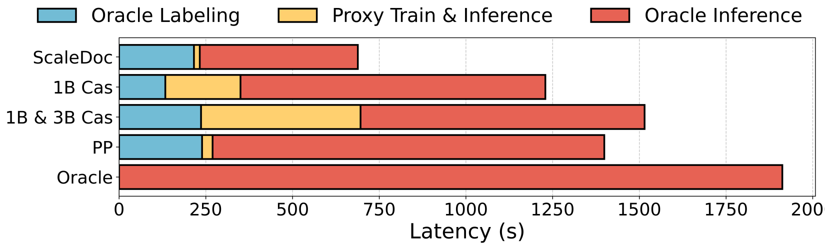


Fig. 5. Breakdown for different approaches over PubMed dataset, measuring average latencies of each stage. – ScaleDoc (top) presents significance improvement.

inference cost of our tiny proxy model; and the substantial reduction in expensive Oracle LLM invocations.

Latency Breakdown. To provide a granular performance analysis, Figure 5 breaks down the average latency of each online processing stage over PubMed datasets. We report 3 different stages of the pipeline, including oracle labeling, proxy train & inference and oracle inference. The oracle-labeling stage represents the overhead of invoking the oracle to label a sample subset for subsequent proxy training and calibration. Compared to other approaches, ScaleDoc achieves significant reduction in oracle inference. Its efficiency is also driven by its lightweight proxy model, which benefits training and inference. In contrast, the LLM cascade methods exhibit a performance bottleneck. Although their proxy LLMs offer training-free zero-shot capabilities, they still incur moderate computational costs during inference, leading to high end-to-end latency.

Offline Overhead. ScaleDoc needs an offline computation for document representation. Encoding 10,000 documents with NvEmbed is a **one-time** process and requires a relatively small amount of computation. For example, on the PubMed dataset, the estimated computation is approximately 50 PFLOPs. Conversely, during online processing, an LLM oracle (e.g., GPT-4o) demands 10× more computation for each ad hoc query. A lightweight proxy model such as Llama-3.2-3B would incur 27 PFLOPS online computation for each new query. This comparison underscores that ScaleDoc's one-time offline overhead is relatively small and acceptable, in light of the efficiency it brings to online query processing.

6.3 Accuracy Analysis

Validation against Human Labels. Since human-labeled data is typically limited to specific classification tasks, our main evaluation relies on various manually curated queries. Nevertheless, to ensure rigorous validation against human standards, we conducted additional experiments using human annotations from the datasets BigPatent and PubMed. We mapped the original classification labels to binary predicates and evaluated the actual end-to-end accuracy of ScaleDoc. Figure 6 demonstrates that our system successfully maintains user-specified accuracy (achieving the *Ideal* dotted line). Besides, across these 11 queries with a target accuracy of $\alpha = 0.9$, ScaleDoc reduces average oracle invocations by 80.6%. These results confirm that our approach is robust against human judgment and maintains high efficiency.

Accuracy Robustness across Selectivities. For predicate evaluation, selectivity denotes the fraction of positive instances, thereby introducing challenges related to class imbalance. To assess robustness against such skewness, Figure 7 reports the accuracy of ScaleDoc across varying selectivities. The results demonstrate that ScaleDoc exhibits resilience to data skew, maintaining

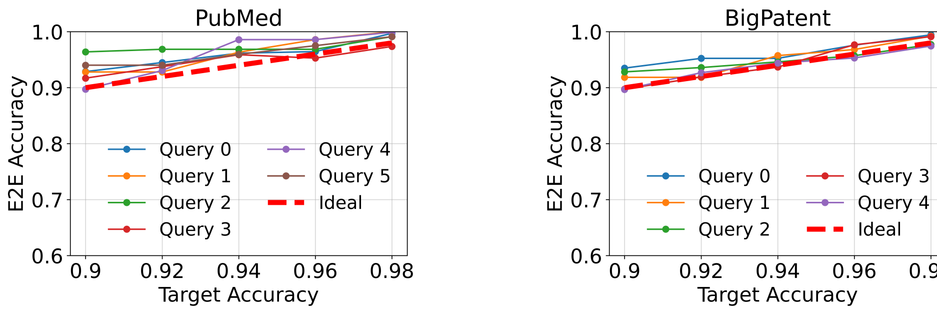


Fig. 6. Accuracy Validation with human-annotated labels.

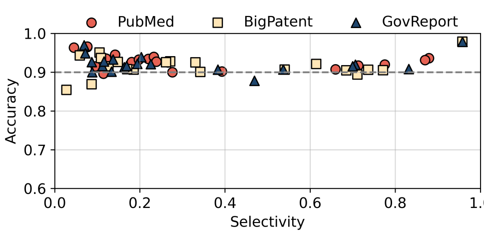


Fig. 7. Accuracy performance over different selectivities.

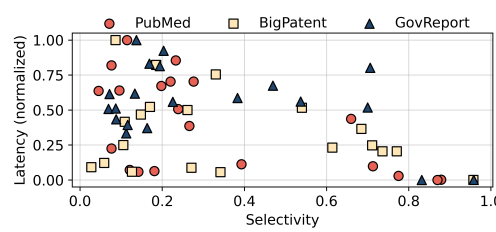


Fig. 8. Latency performance (normalized) over different query selectivity.

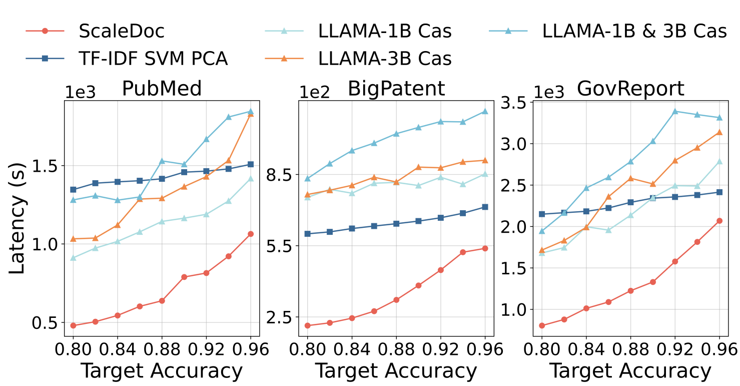


Fig. 9. Accuracy-Latency tradeoff on three datasets.

robust accuracy across a broad range of selectivities (More details about skewness are presented in Section 6.6.)

Accuracy-Latency Tradeoff. We evaluated the accuracy-latency trade-off by setting accuracy targets from 0.80 to 0.96. As shown in Figure 9, ScaleDoc consistently reduces average runtime across three datasets and different accuracy targets. All of the methods exhibit a common trend: higher accuracy targets generally incur more latency. When accuracy targets are relaxed, ScaleDoc's latency decreases by a larger margin, offering more trade-off opportunities for higher performance. In contrast, other baselines show smaller latency variation, due to their unreliable decision scores. This limits their data reduction even when accuracy requirements are loosened.

Table 3. End-to-end processing latencies of two direct embedding-matching methods and ScaleDoc.

	PubMed	BigPatent	GovReport
E5	1412.0	708.1	2257.6
NvEmbed	1358.9	641.2	2206.5
ScaleDoc	**789.1**	**382.6**	**1330.6**

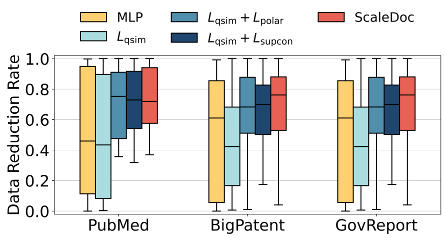


Fig. 10. End-to-end data reduction ablation results on different training loss variants – MLP serves as a baseline, training a standard binary classifier. $\mathcal{L}_{qsim}$, $\mathcal{L}_{supcon}$, $\mathcal{L}_{polar}$ are ablation-style variants of ScaleDoc's training approach.

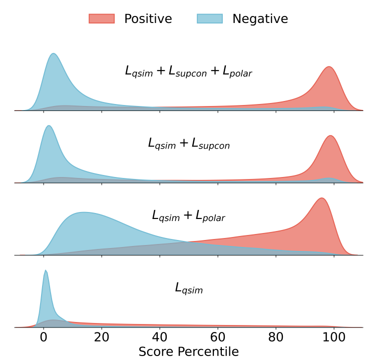


Fig. 11. Average score distribution of positives and negatives – The four model variants are trained with different losses in an ablation style. All scores are normalized into percentiles.

6.4 Ablation for Proxy Model Training

Ablation Comparison. To assess the impact of our overall training framework, we compare it against the baselines that rely on direct embedding-based similarity matching. Specifically, we evaluate on NvEmbed, the embedding model used in ScaleDoc's offline phase, along with a widely adopted model, E5 [45, 46]. For this approach, similarity scores are directly computed between each document and the query, then serving as proxy values for cascading decisions. As shown in Table 3, our query-aware training outperforms these direct approaches. Unlike static latent representations, our paradigm dynamically adapts to online query semantics, producing more reliable decision scores and superior efficiency.

Breakdown of the Training. We further provide a decomposed analysis of the proposed contrastive learning objectives. Figure 10 shows the effectiveness of our training design through end-to-end data reduction results. Here, we use a brute-force optimal cascade to isolate the effect from cascade designs. The key findings include:

1) Superiority of contrastive-based paradigm. We compare ScaleDoc with a baseline where a MLP binary classifier is directly trained on document embeddings. Figure 10 shows that ScaleDoc achieves 20% higher data reduction than basic MLP. Our further investigation suggests that the standard MLP training is inadequate for handling the complex query-document fusion task.

2) Effectiveness of different losses. Our ablation (Figure 10) reveals the distinct roles of each loss. Training with only $\mathcal{L}_{qsim}$ in phase 1 is insufficient, but phase 2, with $\mathcal{L}_{supcon}$ and $\mathcal{L}_{polar}$, resolves this to achieve the final performance. This does not diminish the role of phase 1. In fact, models trained without phase 1 failed to produce valid results, demonstrating that establishing an initial semantic ranking is a fundamental prerequisite.

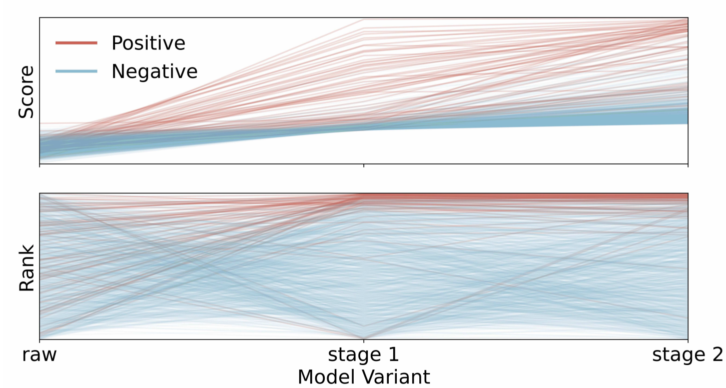


Fig. 12. Embedding relocation mapping during training. – SCORE stands for the numerical values of cosine similarity scores, and RANK stands for the relative order of the scores. Each line demonstrates the relocation trace of a single document in the latent space throughout 2 training phases.

We dive deeper into these objectives with following experiments:

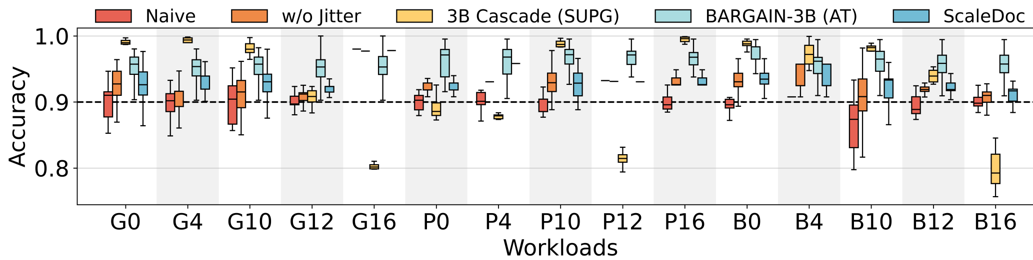
a) $\mathcal{L}_{qsim}$ guarantees semantic monotonicity. Figure 12 illustrates how the lightweight encoder refines document representations. The encoder maps the original embedding (RAW) to a latent space where scores reflect semantic consistency to the predicates. While positive and negative documents initially overlap, stage-1 training with $\mathcal{L}_{qsim}$ makes them semantically orderly: positive documents shift to high-score regions, and negatives to low ones. This establishes *semantic monotonicity*, forming a crucial foundation.

b) $\mathcal{L}_{supcon}$ and $\mathcal{L}_{polar}$ create a bipolar distribution. Figure 11 illustrates score distributions under different training objectives. $\mathcal{L}_{qsim}$ alone yields overlapping positive and negative distributions, complicating threshold selection. Both $\mathcal{L}_{supcon}$ and $\mathcal{L}_{polar}$ address this by creating a *bipolar distribution*, pushing positive and negative scores to opposite ends. Specifically, $\mathcal{L}_{supcon}$ clusters documents with the same label and creates *high-kurtosis peaks*, but it introduces small and incorrect sub-clusters in the tail regions. $\mathcal{L}_{polar}$, in contrast, does not cluster as aggressively, but is effective at shaping the tails. This creates a cleaner separation and mitigates the side effects of $\mathcal{L}_{supcon}$. Integrating all three losses together achieves the reliable final distribution

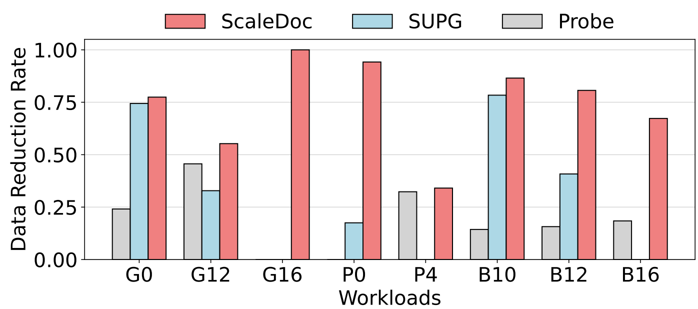
6.5 Validation of the Cascade Mechanism

Accuracy Maintenance. Handling ad hoc cascade is inherently challenging because the system lacks prior knowledge to guarantee specified accuracy targets. To empirically validate the reliability of our cascade accuracy, we compare it against SUPG [19], BARGAIN [48], a *Naive* approach (directly selecting thresholds from the sampled distribution), and an ablation variant of our approach (*w/o Jitter*). We evaluate 100 trials over 15 queries, reporting F1 scores, whereas BARGAIN uses the accuracy ratio. As shown in Figure 13a, the *Naive* approach fails to achieve the target accuracy in nearly half of the trials due to uncontrolled calibration. Similarly, both the *w/o Jitter* variant and SUPG exhibit instability across several queries. Although SUPG leverages statistical bounds for threshold selection, it misses the target in some practical cases due to information bias. Critically, Both BARGAIN and our SCALEDOC maintain a robust accuracy to achieve the specified target. These results confirm SCALEDOC's reliability for accuracy maintenance.

Effect on Data Reduction. Furthermore, we evaluate the effectiveness of different cascade mechanisms in reducing LLM invocation, given the same embedding-based proxy. We introduce



(a) Cascade accuracy on 100 trials – Black dotted line demonstrate accuracy target α=0.90.



(b) Data reduction over different cascade algorithms – SUPG and Probe may result in zero data reduction, represented by blank bars.

Fig. 13. Ad hoc cascade accuracy and data reduction rates.

Table 4. JSD between reconstructed distribution and ground truth (Lower is better). – N denotes Naive stratified sampling. p/n denotes positive and negative distributions.

	N-p	N-n	IS-p	IS-n	B-p	B-n	SD-p	SD-n
Mean	0.30	0.11	0.34	0.18	0.50	0.38	**0.20**	**0.09**
Median	0.32	0.08	0.33	0.17	0.42	0.43	**0.16**	**0.08**

an additional baseline, *Probe-based Calibration*, which iteratively forwards the most ambiguous documents to the oracle (starting from a decision score of 0.5). Figure 13b demonstrates that ScaleDoc delivers superior and generalizable performance. The SUPG and Probing methods struggle with some cases, even yielding zero data reduction. This further validates the effectiveness of our adaptive cascade strategy in maximizing efficiency under accuracy constraints.

Validate Density Estimator. The effectiveness of ad hoc calibration is based on a faithful Density Estimator (DE) for the score distribution. A good DE should accurately model original distributions from a small subset. To validate its effectiveness, we leverage Jenson-Shannon Distance (JSD) to measure the discrepancy between the reconstructed distribution and the original full distribution. We compare the quality of ScaleDoc's DE results (SD) with other estimators, such as Importance Sampling (IS) and directly fitting Beta distribution curves (B). As shown in Table 4, across 10 different queries, our DE (optimized with linear interpolation) achieves the lowest discrepancy in general, ensuring the selected thresholds are accurately transferable to the full set.

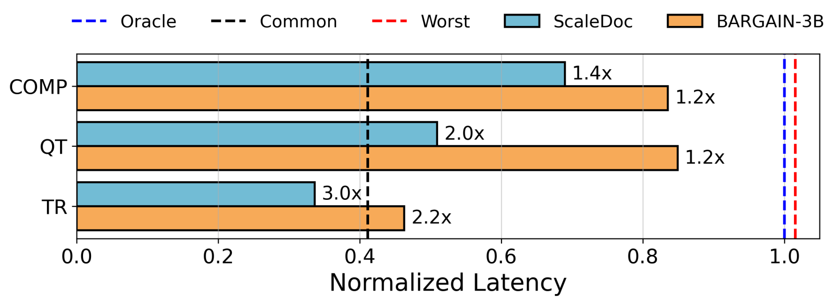


Fig. 14. Normalized latency on extended queries. – We also label the **speed-up ratio** compared to the Oracle. Common denotes ScaleDoc's average latency of existing common queries from PubMed. Worst represents the theoretical worst-case for ScaleDoc (strictly 100% target accuracy, no reduction of Oracle).

6.6 Latency Analysis of the Data Skewness

A key challenge in data predicates is skewness. While Figure 7 shows the relevant accuracy performance, Figure 8 further plots the latency results against selectivity. Queries with higher selectivity achieve greater speedups, as more positive cases form a richer predicate-relevant training set. For low-selectivity queries, performance would be expected to degrade due to the sparse positive data. However, our results reveal a crucial strength: ScaleDoc not only performs well on high-selectivity queries but also delivers evident speedups across low-selectivity scenarios. This resilience stems from our core contributions. First, the contrastive-based training creates a reliable representation space, capturing discriminative semantics even from limited positive samples. Second, our adaptive cascade mechanisms further enhance the robustness across different data distribution.

6.7 Stress-Testing on Complex Queries

With three real-word document collections and carefully curated queries, our main evaluation covers diverse common predicates such as topic classification, standpoint analysis, fact-checking and attribute verification. To further assess robustness beyond them, we stress-test ScaleDoc on more complex queries where embeddings may lack sufficient signals. We categorize these complex queries into three types:

- **Implicit Text Reasoning (TR)**: Queries requiring inference on information not explicitly stated in the text.
  Example. For patent documents, the query could be: "Does this document describe an invention requiring a highly educated person to operate?"
- **Quantitative Analysis (QT):** Tasks with numerical extraction and arithmetic conditions.
  Example: Medical papers often report p-value tests. The query could be: "Does the paper report a p-value less than 0.05?"
- **Composite Predicates (COMP):** Queries combining multiple conditions with logical conjunctions.
  Example: For patent documents, the query could be: "Is this document from a scientific institution and suitable for common usage in daily life?"

We curate 10 new queries across these categories and set the target accuracy as 0.9. As shown in Figure 14, ScaleDoc consistently outperforms Oracle and BARGAIN-3B, achieving 1.4×–3.0× speedups. While effective, our speedups for Quantitative Analysis and Composite Predicates are

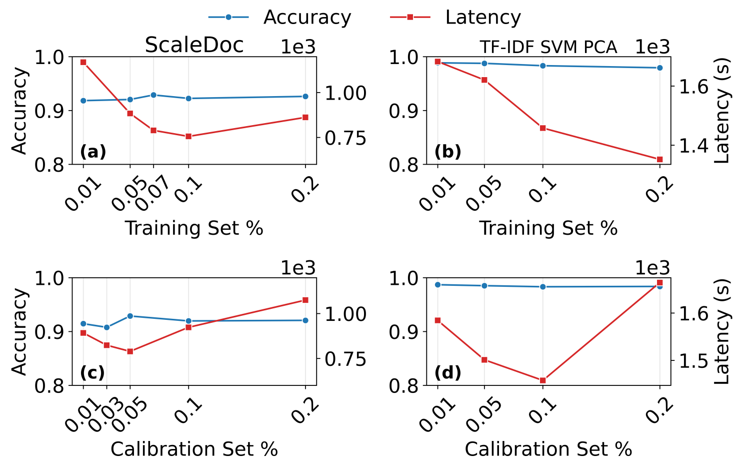


Fig. 15. Accuracy and Latency with different hyperparameters – Training Set % and Calibration Set % denotes the portion of data sampled from the global.

lower compared to other queries. This confirms that semantic embeddings are less effective in capturing symbolic patterns (e.g., numerical values) and complex logic. However, our system could still mitigate this and achieve speedups with adaptive training and effective cascade. Even in the theoretical worst case where the proxy filters no documents, the overhead of our lightweight proxy is negligible, compared to the expensive LLM inference. This further ensures that ScaleDoc remains robust in complex scenarios.

6.8 Impact of Hyperparameters

We demonstrate the impact of different hyperparameters, the sizes of training and calibration sets, on ScaleDoc's end-to-end performance with a target accuracy of 0.90 (see Figure 15). For the training set, larger sizes initially improve the proxy model's ability to learn global features, enhancing efficacy. However, this eventually led to increased labeling overhead and higher latency. The preferred trade-off is empirically found with a training set size between 7% and 10%. Conversely, a traditional PP requires much more labels to train, leading to more LLM oracle invocations. A similar trade-off exists for the calibration set, where sampling 5% is sufficient to capture global distributions, maximizing end-to-end performance.

7 Related Work

LLMs in Data Systems. Large Language Models (LLMs) has catalyzed a paradigm shift in data analysis, with their remarkable zero-shot capabilities. For instance, numerous studies have explored using LLMs to translate natural language into SQL queries [25, 28, 30], lowering the barrier to database analytics. LLMs are also integrated as agentic components to orchestrate complex data processing workflows [14, 40, 49]. Furthermore, analytical systems such as LOTUS [34] and Palimpzest [27] formulate *semantic operators*, which provide and integrate several declarative LLM interfaces over unstructured data. Other works target various essential tasks such as multiclass text classification [21], data cleaning [50], and entity extraction [11]. In parallel, ScaleDoc addresses the foundational and ubiquitous task of semantic predicating. Our goal is to develop a general, scalable, and cost-effective solution for this primitive, which is critical for a wide range of analytical scenarios.

Acceleration for Machine Learning Queries. The high computational cost of machine learning (ML) inference, particularly for LLMs, has made acceleration a critical area of research. Some works focus on hardware optimizations. For instance, PagedAttention [23] and FlashAttention [6] improve performance by optimizing the attention mechanism and memory access at the GPU level. Other scheduling approaches, such as DistServe [52] and Sarathi [1], optimize throughput via disaggregated phases or chunked prefilling. Our work is orthogonal to these approaches, focusing on a higher layer of the analytical system stack.

Additionally, KV caching mechanisms for LLMs are cost-effective by reusing shared prefixes, which is common in multi-turn dialogues or specific context comprehension [8, 51]. However, our workload requires traversing an enormous and diverse set of documents. Since these documents are distinct and lack significant shared prefixes, the overall cache reuse is negligible. Besides, extending the KV cache to cover the entire document collection would incur prohibitive memory overhead, rendering this approach infeasible for our use case.

At the higher system level, some recent works adopt the proxy-cascades workflow. In this paradigm, a lightweight proxy model rapidly filters out easy entities before sending them to a powerful, expensive model. For instance, Probabilistic Predicates (PPs) [29, 47] employed simple classifiers as pre-filtering proxies, and NoScope [18] built a cascade video databases to accelerate object detection. However, these approaches lack generalizability to new workload patterns, requiring labor-intensive manual adaptation. More recently, systems such as FrugalGPT [4] and Palimpzest [27] use smaller LLMs as proxies for more powerful ones (e.g, GPT-4o). But these solutions still depend on the moderate-scale LLM proxies, remaining expensive. Furthermore, given the ad hoc and diverse workload, directly fine-tuning small LLMs (e.g., 7B-parameter models) as the oracle is infeasible, due to the substantial computational overhead of online training [44]. Our work addresses these limitations by exploring a more generalizable, lightweight, and automated solution for semantic queries.

Semantic Representation. Semantic representations encode the meaning of text into dense vectors [36]. Early advances, such as Sentence-BERT [37] and SimCSE [9], established the effectiveness of embeddings for semantic tasks. More recently, the focus has shifted to leveraging LLMs to generate high-quality, context-aware embeddings [3, 24, 26]. These models have demonstrated state-of-the-art performance in various downstream applications [31, 42]. Our work does not propose a new representation model. Instead, we focus on the systems challenge of efficient workflow orchestration and treat the representation model as a pluggable component.

8 Conclusion

In this paper, we introduce ScaleDoc, a novel system designed to efficiently scale LLM-based semantic predicates over large document collections. It decouples the execution into an offline representation phase and a highly optimized online filtering phase. The system's effectiveness stems from two key innovations: (1) a query-aware proxy model to produce reliable predicating scores, and (2) an ad hoc cascade workflow that dynamically determines the data filtering with specified accuracy targets. Experiments on diverse datasets confirm our design, showing ScaleDoc reduces LLM calls by up to 85% and accelerates end-to-end performance by more than 2×. Our work highlights the potential for the use of LLMs in large-scale data analysis systems with effective performance.

Acknowledgments

This paper was supported by the National Science Foundation of China (Grant No. 62532001), Xiongan AI Institute, and Shanghai Qi Zhi Institute.

References

[1] Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav Gulavani, Alexey Tumanov, and Ramachandran Ramjee. 2024. Taming {Throughput-Latency} tradeoff in {LLM} inference with {Sarathi-Serve}. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*. 117–134.

[2] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes. *Proceedings of the VLDB Endowment* 17, 2 (2023), 92–105.

[3] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. In *First Conference on Language Modeling*. https://openreview.net/forum?id=IW1PR7vEBf

[4] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv:2305.05176 [cs.LG] https://arxiv.org/abs/2305.05176

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PmLR, 1597–1607.

[6] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems* 35 (2022), 16344–16359.

[7] Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Greg Kondrak and Taro Watanabe (Eds.). Asian Federation of Natural Language Processing, Taipei, Taiwan, 308–313. https://aclanthology.org/I17-2052/

[8] Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. 2024. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *arXiv preprint arXiv:2407.11550* (2024).

[9] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6894–6910. doi:10.18653/v1/2021.emnlp-main.552

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[11] Chuxuan Hu, Austin Peters, and Daniel Kang. 2025. LEAP: LLM-powered End-to-end Automatic Library for Processing Social Science Queries on Unstructured Data. *arXiv preprint arXiv:2501.03892* (2025).

[12] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[13] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient Attentions for Long Document Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 1419–1436. doi:10.18653/v1/2021.naacl-main.112

[14] Yulong Hui, Chao Chen, Zhihang Fu, Yihao Liu, Jieping Ye, and Huanchen Zhang. 2025. Interact-RAG: Reason and Interact with the Corpus, Beyond Black-Box Retrieval. *arXiv preprint arXiv:2510.27566* (2025).

[15] Yulong Hui, Yihao Liu, Yao Lu, and Huanchen Zhang. 2025. Okralong: A flexible retrieval-augmented framework for long-text query processing. *arXiv preprint arXiv:2503.02603* (2025).

[16] Yulong Hui, Yao Lu, and Huanchen Zhang. [n. d.]. UDA: A Benchmark Suite for Retrieval Augmented Generation in Real-World Document Analysis. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[17] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[18] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Neural Network Queries over Video at Scale. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[19] Daniel Kang, Edward Gan, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2020. Approximate selection with guarantees using proxies. *Proc. VLDB Endow.* 13, 12 (July 2020), 1990–2003. doi:10.14778/3407790.3407804

[20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. [n. d.]. Dense Passage Retrieval for Open-Domain Question Answering.

[21] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2024. Chorus: Foundation Models for Unified Data Discovery and Exploration. *Proceedings of the VLDB Endowment* 17, 8 (2024), 2104–2114.

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems* 33 (2020), 18661–18673.

[23] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*. 611–626.

[24] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428* (2024).

[25] Zhenwen Li and Tao Xie. 2024. Using LLM to select the right SQL Query from candidates. *arXiv preprint arXiv:2401.02115* (2024).

[26] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[27] Chunwei Liu, Matthew Russo, Michael Cafarella, Lei Cao, Peter Baille Chen, Zui Chen, Michael Franklin, Tim Kraska, Samuel Madden, and Gerardo Vitagliano. 2024. A declarative system for optimizing ai workloads. *arXiv preprint arXiv:2405.14696* (2024).

[28] Shicheng Liu, Jialiang Xu, Wesley Tjangnaka, Sina Semnani, Chen Yu, and Monica Lam. 2024. SUQL: Conversational Search over Structured and Unstructured Data with Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 4535–4555. doi:10.18653/v1/2024.findings-naacl.283

[29] Yao Lu, Aakanksha Chowdhery, Srikanth Kandula, and Surajit Chaudhuri. 2018. Accelerating machine learning inference with probabilistic predicates. In *Proceedings of the 2018 International Conference on Management of Data*. 1493–1508.

[30] Kyle Luoma and Arun Kumar. 2025. SNAILS: Schema Naming Assessments for Improved LLM-Based SQL Inference. *Proceedings of the ACM on Management of Data* 3, 1 (2025), 1–26.

[31] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. 2022. Large Dual Encoders Are Generalizable Retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 9844–9855.

[32] OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, and Andrew Codispoti. 2024. GPT-4o System Card. arXiv:2410.21276 [cs.CL] https://arxiv.org/abs/2410.21276

[33] Liana Patel, Siddharth Jha, Carlos Guestrin, and Matei Zaharia. 2024. Lotus: Enabling semantic queries with llms over tables of unstructured and structured data. *arXiv preprint arXiv:2407.11418* (2024).

[34] Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. 2025. Semantic Operators and Their Optimization: Enabling LLM-Based Data Processing with Accuracy Guarantees in LOTUS. *Proc. VLDB Endow.* 18, 11 (July 2025), 4171–4184. doi:10.14778/3749646.3749685

[35] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[36] Abhinav Ramesh Kashyap, Thanh-Tung Nguyen, Viktor Schlegel, Stefan Winkler, See-Kiong Ng, and Soujanya Poria. 2024. A Comprehensive Survey of Sentence Representations: From the BERT Epoch to the CHATGPT Era and Beyond. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 1738–1751. https://aclanthology.org/2024.eacl-long.104/

[37] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

[38] Ricardo Salazar-Díaz, Boris Glavic, and Tilmann Rabl. 2024. Inferdb: In-database machine learning inference using indexes. *Proceedings of the VLDB Endowment* 17, 8 (2024), 1830–1842.

[39] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[40] Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G Parameswaran, and Eugene Wu. 2024. DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing. *arXiv preprint arXiv:2410.12189* (2024).

[41] Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 2204–2213. doi:10.18653/v1/P19-1212

[42] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Annual Meeting of the Association for Computational Linguistics-ACL 2023 (09/07/2023-14/07/2023,,, Toronto, Canada)*.

[43] The Llama 3 Team, Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, and Chris McConnell. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[44] Kushala VM, Harikrishna Warrier, Yogesh Gupta, et al. 2024. Fine tuning llm for enterprise: Practical guidelines and recommendations. *arXiv preprint arXiv:2404.10779* (2024).

[45] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. arXiv:2212.03533 [cs.CL] https://arxiv.org/abs/2212.03533

[46] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. arXiv:2402.05672 [cs.CL] https://arxiv.org/abs/2402.05672

[47] Zhihui Yang, Zuozhi Wang, Yicong Huang, Yao Lu, Chen Li, and X Sean Wang. 2022. Optimizing machine learning inference queries with correlative proxy models. *Proceedings of the VLDB Endowment* 15, 10 (2022), 2032–2044.

[48] Sepanta Zeighami, Shreya Shankar, and Aditya Parameswaran. 2025. Cut Costs, Not Accuracy: LLM-Powered Data Processing with Guarantees. *Proc. ACM Manag. Data* 3, 6, Article 311 (Dec. 2025), 26 pages. doi:10.1145/3769776

[49] Enhao Zhang, Nicole Sullivan, Brandon Haynes, Ranjay Krishna, and Magdalena Balazinska. 2025. Self-Enhancing Video Data Management System for Compositional Events with Large Language Models. *Proc. ACM Manag. Data* 3, 3, Article 215 (June 2025), 29 pages. doi:10.1145/3725352

[50] Shuo Zhang, Zezhou Huang, and Eugene Wu. 2024. Data cleaning using large language models. *arXiv preprint arXiv:2410.15547* (2024).

[51] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. 2024. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems* 37 (2024), 62557–62583.

[52] Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving. In *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation* (Santa Clara, CA, USA) *(OSDI'24)*. USENIX Association, USA, Article 11, 18 pages.

Received October 2025; revised January 2026; accepted February 2026