Iterative Refinement of LLM as Judge Framework

Rahul Sengupta rasengupta@ucsd.edu
Zachary Thomason zthomason@ucsd.edu
Zeyu (Edward) Qi zeqi@ucsd.edu
Akshay Medidi amedidi@ucsd.edu
Ryan Lingo ryanlingo@gmail.com

Abstract

Large Language Models (LLMs) can produce coherent and informative summaries of complex lecture materials, but evaluating those summaries at scale raises critical issues. Human review is costly and subjective, while common automatic metrics (e.g., ROUGE, BLEU, BERTScore) often fail to capture deeper semantic understanding and factual consistency. To address this, we developed an evaluation framework that uses GPT-5 both to generate lecture-slide summaries and to assess them using a rubric paired with task-based benchmarks, including coverage, faithfulness, organization, clarity, and style. We track performance through dashboards and validate alignment with human standards through specific quality signals. Building on this baseline, we are expanding the system to become a more adaptive, reference-free evaluator. A judge module first pushes each lecture to a domain-aware rubric drawn from a data bank, enabling evaluation criteria, and even the number of refinement iterations, to vary based on lecture domain and summary type. We also integrate a Chain-of-Density refinement stage to produce more comprehensive and information-dense summaries, and adjust scoring outputs to better reflect domain-based quality flags across iterations. Overall, we aim to improve the reliability and scalability of LLM-as-judge evaluation by combining deterministic benchmarks with domain-adaptive rubrics and controlled iterative refinement.

Code: https://github.com/rahul-sg/DSC180A-Final-Project-Honda

1 Introduction

Large language models (LLMs) are rapidly expanding what is possible in natural language processing. They can generate fluent text across a wide range of tasks, but as these systems become more advanced, evaluating their outputs in a way that is systematic, meaningful, and interpretable becomes increasingly difficult. Traditional NLP metrics such as ROUGE, BLEU, METEOR, and embedding-based measures like BERTScore can be useful quality indicators, but they often overvalue similarity and fail to capture practical relevance. Important qualities like factual grounding, semantic completeness, domain-appropriate structure, and overall usefulness to human readers are not reflected through a calculated score. This motivates the central goal of our project, which is to build an iterative evaluation system that combines LLM-as-judge scoring with human-guided rubrics and task-based NLP metrics to produce a scalable evaluator that aligns closely with human judgment. We study this problem through lecture slide summarization. Lecture slides are a controllable input, which makes it possible to compare evaluation strategies consistently and to isolate where different methods succeed or fail. It is also a very relevant metric to us as students in academia. Our objective is not to optimize summarization, but to design an evaluation framework that can meaningfully assess a wide range of generated outputs. In particular, we focus on LLM-as-judge evaluation, where an LLM is prompted with a rubric to assess another model's summary. This approach is flexible and scalable, but is also sensitive to prompt design and criteria definition. To make LLM-as-judge evaluation more reliable, we pair rubric-based judging with task-based checks that provide more deterministic signals, such as coverage and grounding measures. We also move beyond static rubrics by introducing domain-aware criteria that reflect what different types of lectures value. Instead of forcing the same dimensions onto every summary, our evaluator selects from a structured bank of domain rubrics so that the scoring dimensions and feedback are better aligned with the content. Finally, we incorporate an iterative refinement process that uses evaluation feedback to target weaknesses in a summary and improve it across iterations. Together, these components aim to support evaluation that is scalable, better grounded in the source material, and more aligned with human standards across diverse lecture domains.

1.1 Literature Review

Evaluation of LLM-generated text is a topic that is becoming increasingly important, but it remains complex because generated text quality is not easily assessed with a singular, agreed-upon metric. When it comes to evaluating generated summaries, research often frames evaluation as a collection of different goals rather than a single score. These goals include whether a summary is faithful to the source, whether it captures the most important ideas, and how efficient it is to produce, among other things. Because these dimensions can matter differently depending on the context, the literature generally suggests using multiple complementary measures instead of relying on one metric to represent "quality."

Traditional automated metrics are widely used because they are efficient, reproducible, and easy to scale. Metrics such as BLEU and ROUGE can be useful for benchmarking, but

they largely focus on surface-level similarity between a generated output and a reference. ROUGE in particular evaluates summaries through overlap-based statistics, including n-gram overlap and variants based on longest common subsequence, which makes it effective for measuring similarity to a reference summary when strong references exist. However, these overlap-based methods are limited in their ability to evaluate semantic understanding, grounding, and whether the model is aligning with a human's preferences. As a result, automated metrics can provide helpful flags but often fail to capture the full range of qualities that humans care about when reading a summary.

Human evaluation is the gold standard when it comes to evaluation because humans can directly judge factors like meaning, context, tone, and usefulness in ways that automated metrics do not. However, literature highlights that human evaluation is expensive and time-consuming, especially when applied across many examples and multiple criteria. Human ratings are also inherently subjective and can vary across evaluators, which creates challenges in consistency and reproducibility. This makes human judgment valuable for validation, but difficult to rely on as the primary evaluation method for large-scale systems.

LLM-as-judge evaluation has emerged as a scalable alternative that can apply rubric-based criteria to generated text and provide structured feedback without requiring a reference to build upon. This approach is attractive because it is flexible and can be tailored to specific evaluation goals through careful prompting choices. That being said, prior work also shows that LLM-as-judge systems can be unstable and sensitive, and that evaluations can diverge from human ratings in cases that require deeper reasoning. Recent research explores ways to improve reliability through iterative workflows, including refining evaluation prompts with human feedback and expanding coverage using synthetic test cases that expose edge cases. Overall, the literature points toward hybrid strategies that combine the scalability of LLM judging with more deterministic task-based signals and human sanity checks.

1.2 Discussion of Prior Work

Across these evaluation strategies, the main pattern present is the tradeoff between scalability and reliability. Reference-based metrics scale efficiently and provide quality signals, but they do not reliably reflect human wants in a strong summary. Human evaluation captures these qualities more directly, but it is not realistic to apply broadly at scale and introduces variance from subjective judgment. LLM-as-judge methods sit in the middle and can evaluate richer criteria, but they introduce their own reliability risks and do not automatically align with human standards, especially for more complex content like that present in academia.

This motivates our attempt at a combined approach rather than treating any single method as sufficient. Deterministic task-based metrics can serve as consistent checks for measurable behaviors and critical failures, while rubric-driven LLM judging can evaluate higher-level qualities that deterministic metrics cannot represent. Prior work also suggests that LLM judging becomes more useful when it is guided through iteration and human supervision. Rather than assuming the judge's scores are automatically reliable, we treat them as a component with tunable levers whose behavior improves through targeted testing and feedback.

Another important implication is that evaluation criteria should match requirements for the type of content being summarized. What counts as a "good" summary is not uniform across domains, and a fixed rubric can miss key aspects depending on the lecture topic. This strengthens the case for adaptive evaluation, where the rubric and feedback better reflect what the lecture domain values. Taken together, the literature supports building an iterative judge system that integrates human-guided rubrics with LLM-based evaluation and task-specific metrics, with the goal of producing an evaluator that remains scalable while aligning with human judgment and providing feedback about why a summary was scored the way it was.

2 Methods

2.1 Overview of the Evaluation Pipeline

Our system evaluates and improves LLM-generated lecture summaries through a reference-free, end-to-end pipeline with four stages: (1) slide text extraction and preprocessing, (2) initial summary generation, (3) iterative judge-refine-select cycles with trend-aware stopping, and (4) hybrid final scoring with reproducible artifact logging.

Given a lecture slide deck, the pipeline first generates or reuses an initial summary (S_0), then refines it over multiple iterations through a judge-refiner loop guided by rubric feedback, pairwise candidate selection, and a lever-based stopping controller. At finalization, the system applies a best-of-last-k safeguard ($k = 3$ in the current implementation) to reduce end-of-run noise from a weak final rewrite. A final quality score is then computed by combining domain-aware rubric evaluation with deterministic alignment signals derived directly from the slide content, and the run is saved as structured intermediate and final artifacts for downstream analysis.

2.2 Text Extraction and Preprocessing

The first stage of our pipeline extracts textual content from lecture slides in PDF format. For PDF documents, we utilize PyMuPDF to extract text page-by-page. Lecture content is loaded from PDF slides and represented as structured slide units (e.g., title and content fields). To reduce noise, we filter out non-content slides (such as logistics-heavy slides) before computing deterministic metrics. We also derive section-level and glossary-level lexical features from slide text to support coverage and hallucination diagnostics during evaluation.

2.3 Iterative Summary Generation

2.3.1 Motivation from Chain of Density

Our iterative refinement approach draws significant inspiration from the Chain of Density (CoD) method introduced by Adams. The CoD technique addresses a fundamental challenge in summarization: determining the optimal amount of information to include in a summary while maintaining readability. As Adams et al. note, "selecting the 'right' amount of information to include in a summary is a difficult task. A good summary should be detailed and entity-centric without being overly dense and hard to follow"(Adams et al).

The original CoD prompt operates through an iterative mechanism that generates increasingly dense summaries. Specifically, the method:

1. Identifies 1-3 informative entities from the source text that are missing from the previous summary
2. Writes a new, denser summary of identical length which covers every entity and detail from the previous summary plus the missing entities

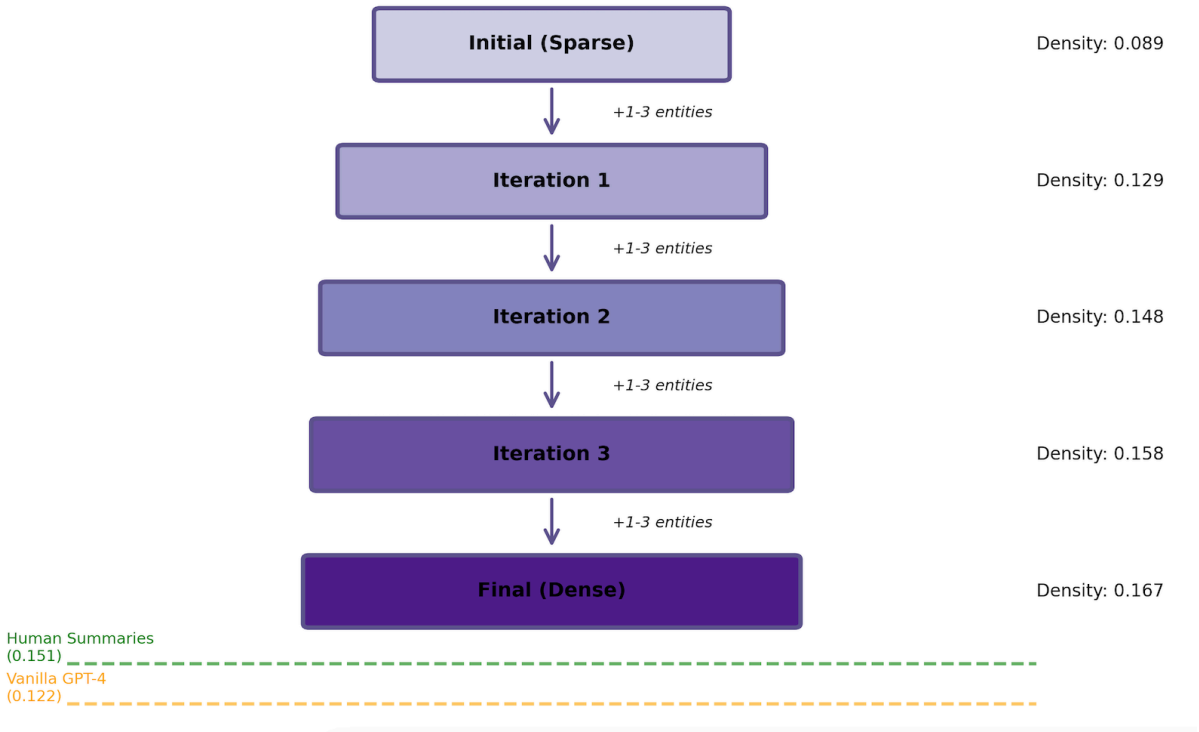

Figure 1: Chain of Density Concept (Adams et al., 2023)

Figure 1 illustrates the Chain of Density progression, showing how entity density increases from 0.089 (initial sparse summary) to 0.167 (final dense summary) over 5 iterations while maintaining fixed length.

The CoD method employs several key principles that inform our approach:

Fixed-Length Constraint: The CoD prompt largely adheres to a fixed token budget, ensuring that increased density comes from better abstraction rather than simply adding more text.

Entity-Based Densification: Adams et al. report that "entity density rises—starting at 0.089, initially below Human and Vanilla GPT-4 (0.151 and 0.122)—to 0.167 after 5 steps of densification"(Adams et al).

Iterative Refinement: The process makes summaries increasingly concise through fusion, compression, and removal of uninformative phrases.

Critically, Adams found that humans prefer summaries that are almost as dense as human-written summaries, with an optimal entity density around 0.15 entities per token (Adams et al.)

2.3.2 Initial Summary Generation

For each lecture, an initial summary S_0 is generated by prompting `gpt-5-chat-latest` with the filtered slide content. The prompt instructs the model to produce a concise lecture summary that captures the major topics, preserves factual accuracy, and avoids unsupported claims. If a valid cached S_0 already exists on disk, it is reused rather than regenerated in order to improve reproducibility and reduce API-driven variance across experiment runs.

2.3.3 Iterative Refinement with Pairwise Selection

Starting from S_0, the pipeline runs iterative judge-refine-select cycles as described in Algorithm 1. At each iteration t, an ensemble judge scores the current summary on five *rubric levers*: coverage, faithfulness, organization, clarity, and style (each rated 1–5). We use the term *lever* to refer to each rubric dimension because the refiner is guided to improve the weakest-scoring dimensions while preserving dimensions that are already strong.

The refiner then proposes a candidate summary $\hat{S}_t$ conditioned on the slide content and structured judge feedback. A pairwise preference judge compares the current and candidate summaries holistically and selects the stronger one as input to the next iteration. This prevents a low-quality single rewrite from propagating and helps stabilize quality across iterations.

At each iteration, the judge provides both numeric rubric scores and qualitative critiques. The refiner proposes an improved candidate, and pairwise selection determines whether the candidate should replace the current summary. Stopping is controlled by a trend-aware decision table, and at finalization the pipeline applies a best-of-last-k safeguard ($k = 3$ in the current implementation) to reduce end-of-run noise from a single weak rewrite.

At each iteration, the judge provides both numeric rubric scores and qualitative critiques. The refiner proposes an improved candidate, and pairwise selection determines whether

Algorithm Iterative Refinement with Lever-Based Guidance and Optional Pairwise Selection

Input: Lecture slides L, initial summary S_0
Output: Final summary S^*
Save S_0 as `iter_0.txt`
$S \leftarrow S_0$
$\mathscr{H} \leftarrow [\,]$
for $t = 1$ to $T_{\max}$ **do**
  $F_t \leftarrow \text{JudgeRubric}(L, S)$
  Compute deterministic signals, change magnitude, and composite quality score
  Append current state $(t, S, F_t, \text{signals}, \text{quality})$ to $\mathscr{H}$
  if $\text{ShouldStop}(t, \text{rubric}, \text{signals}, \text{trend})$ **then**
    break
  end if
  $\hat{S}_t \leftarrow \text{Refine}(L, S, F_t)$
  if `use_pairwise` **then**
    $S \leftarrow \text{PairwiseSelect}(L, \{S, \hat{S}_t\})$
  else
    $S \leftarrow \hat{S}_t$
  end if
  Save S as `iter_t.txt`
end for
$S^* \leftarrow \text{BestOfLastK}(\mathscr{H}, k = 3)$
Save S^* as `final.txt`
Save full metrics and metadata to `result.json`
return $S^* = 0$

the candidate should replace the current summary. This mechanism prevents low-quality rewrites from propagating and empirically stabilizes improvement across iterations.

2.4 Lever-Based Dynamic Stopping

Rather than running a fixed number of refinement rounds, we use a `LeverBasedRefinementController` that applies a decision table over multiple stopping signals. A minimum of 4 iterations is always enforced before any stopping condition is evaluated, while a hard maximum of 12 prevents runaway loops.

The controller tracks four signals at each iteration:

- **Quality score**: a composite $[0, 1]$ score computed upstream from rubric and signal information.
- **Deterministic signals**: length error, section coverage, glossary recall, and suspected hallucination rate (defined in Section2.5).
- **Change magnitude**: Jaccard-based token-level dissimilarity between consecutive summaries.
- **Quality and signal history**: rolling windows of the above metrics over the last 3 iterations.

Stopping is triggered by one of five conditions, checked in the following order:

1. **Pass**: either (a) all signal thresholds are met, word count is within ±15% of the 350-word target, and quality score ≥ 0.74, or (b) the quality score is high and has plateaued. For iterations 1–3, the controller uses relaxed signal thresholds (length error ≤ 0.15, section coverage ≥ 0.85, glossary recall ≥ 0.55, hallucination rate < 0.55); after iteration 3, it switches to a stricter threshold set (length error ≤ 0.12, section coverage ≥ 0.88, glossary recall ≥ 0.62, hallucination rate < 0.48).
2. **Borderline**: quality score ≥ 0.68 and the quality trajectory has plateaued, with signal plateau also required when signal history is available.
3. **Stalled**: hallucination rate, glossary recall, and quality score have not improved meaningfully over the recent trend window (below thresholds of 0.02, 0.02, and 0.015, respectively).
4. **Converged**: after the early iterations, change magnitude falls below 0.03, indicating that consecutive summaries are nearly identical.
5. **Max_iters**: the refinement loop reaches 12 iterations without satisfying an earlier stopping condition.

2.5 Deterministic Signal Metrics

We compute four deterministic signals from lexical alignment between the summary and the filtered slide content. These signals are model-free and serve as interpretable diagnostics during refinement and as inputs to the stopping controller.

1. **Length error**: $|w_s - w^*|/w^*$, where w_s is the summary word count and $w^* = 350$ is

the target length used by the refinement controller.

2. **Section coverage**: the fraction of slide sections for which at least one high-salience keyword, extracted from section title and content, appears in the summary.
3. **Glossary recall**: the proportion of extracted lecture terms that appear at least once in the summary.
4. **Suspected hallucination rate**: the fraction of summary sentences whose lexical similarity to all slide sentences remains below a fixed threshold, implemented through TF-IDF sentence matching.

2.6 Hybrid Final Scoring

The final score combines two complementary scoring components, both derived from the ensemble rubric output and deterministic signals.

The *comprehensive score* $C \in [0,1]$ is a domain-aware weighted average of the five rubric dimensions, normalized by the maximum achievable score:

$$C = \frac{1.0 \cdot \text{coverage} + 2.5 \cdot \text{faithfulness} + 1.0 \cdot \text{organization} + 1.0 \cdot \text{clarity} + 0.5 \cdot \text{style}}{6.0 \times 5}$$

The *manual weighted score* M blends a rubric baseline with the deterministic section coverage signal:

$$\text{base} = \frac{1 \cdot \text{coverage} + 2 \cdot \text{faithfulness} + 1 \cdot \text{organization} + 1 \cdot \text{clarity} + 1 \cdot \text{style}}{6 \times 5}$$

$$M = 0.8 \cdot \text{base} + 0.2 \cdot \text{section\_coverage}$$

The two components are blended into a pre-damping score:

$$S = 0.7C + 0.3M$$

A domain-aware hallucination damping factor γ is then applied to compute the final score:

$$\alpha_{\text{eff}} = 0.05 \cdot \delta_d$$

$$\gamma = \text{clamp}(1 - \alpha_{\text{eff}} \cdot h,\ 0.75,\ 1.0)$$

$$S_{\text{final}} = \gamma \cdot S$$

where $h \in [0,1]$ is the suspected hallucination rate and δ_d is a domain multiplier that reflects the stricter factual expectations of technical fields. The supported values are shown in Table1.

Together, C and M differ in how they weight faithfulness (2.5× vs. 2×) and whether section coverage is explicitly incorporated, providing two complementary views of quality that the blend regularizes. The clamp floor of 0.75 ensures that even high hallucination rates do not catastrophically suppress otherwise high-quality summaries, while the domain multiplier δ_d imposes heavier penalties in technical domains where factual precision is more critical.

Table: Domain-aware hallucination penalty multipliers δ_d.

Domain	δ_d
Engineering	1.15
Mathematics	1.10
Natural Sciences	1.05
Social Sciences	1.00
Business	0.95
Humanities	0.90

2.7 Implementation Details

2.7.1 Model Configuration

Our implementation uses configurable model endpoints through a shared `LLMConfig` interface. In the main experiments reported in this paper, the summarization, judging, and refinement roles are all instantiated with `gpt-5-chat-latest`, with role-specific completion limits. The configuration interface also supports optional temperature and seed settings when needed for controlled experiments.

In the final evaluation layer, rubric scores are aggregated across three independent judge runs for stability. Within the iterative refinement loop, however, the stopping controller uses the single-run judge path rather than the ensemble scorer.

2.7.2 Chunking Strategy for Long Lectures

To handle lectures that may exceed model context windows, we implement token-based chunking while preserving slide boundaries. Tokens are estimated using a 4-characters-per-token heuristic:

$$\text{tokens}_{\text{est}}(t) = \max\left(1, \left\lfloor \frac{|t|}{4} \right\rfloor\right) \tag{1}$$

where $|t|$ is the character length of text t. Slides are then grouped into bounded chunks, which are later converted into structured text blocks for prompting.

2.8 Comparison with Prior Work

Table2 compares our implementation with the Chain of Density approach. While our system adopts the iterative refinement concept, it differs in several key aspects.

Figure2 illustrates the relationship between entity density and quality metrics across different approaches, highlighting the optimal density range identified by Adams.

Table: Feature comparison with Chain of Density

Feature	Chain of Density	Our System
Iterative refinement	✓	✓
Fixed-length constraint	✓	×
Entity tracking	✓	×
Pairwise selection	×	✓
Adaptive stopping	×	✓
Best-of-last-k safeguard	×	✓
Deterministic signals	Limited	Comprehensive
Reference-free default evaluation	×	✓
Target density enforced	✓	×
Iterations	5 fixed	4–12 adaptive
Domain	News articles	Lecture slides

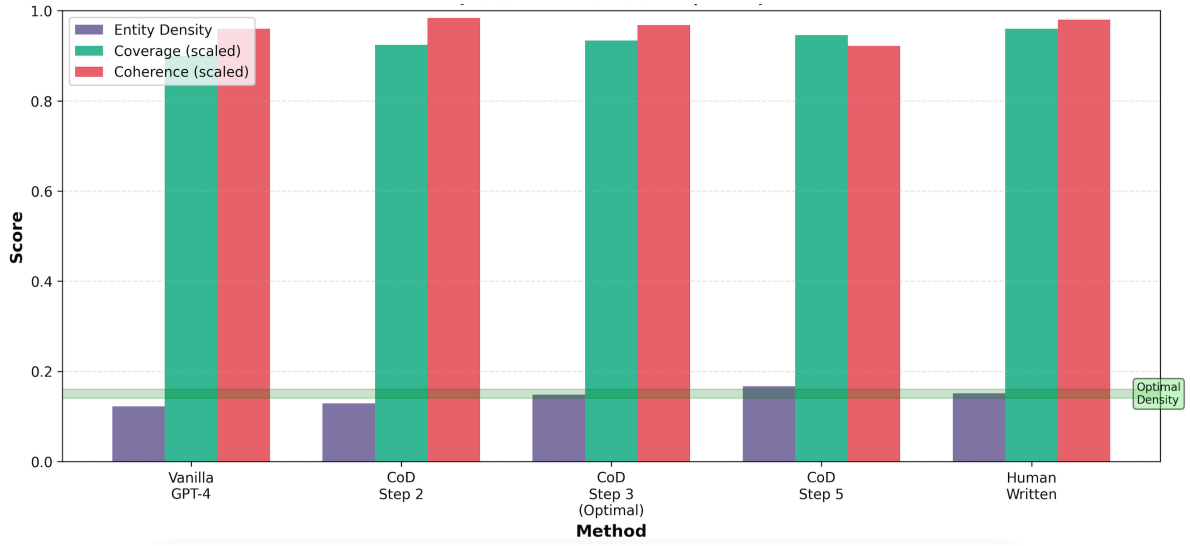

Figure: Comparison of summarization approaches based on Adams et al.

3 Experiments and Results

3.1 Dataset

We evaluate our pipeline on seven lecture slide decks spanning five detected domains: business (Lectures 1–2), humanities (Lecture 3), natural sciences (Lectures 4 and 7), social sciences (Lecture 5), and engineering (Lecture 6). Topics include introductory accounting, double-entry bookkeeping and financial statements, modernity and decoloniality, evolutionary genetics / human origins, the social construction of race, SQL subqueries, and within-subjects experimental design. All slides are provided as PDF files and processed by

the same extraction and filtering pipeline described in Section2.

3.2 Experimental Setup

Our primary experiment evaluates the current pipeline in its default **reference-free** setting. For each lecture, the system generates or reuses an initial summary S_0, applies iterative refinement with pairwise candidate selection, and computes a final risk-adjusted score using the tuned hybrid scoring policy described in Section2.6. In this setting, the raw quality score is

$$Q_{\text{raw}} = 0.7C + 0.3M,$$

and the final reported score is

$$Q_{\text{risk}} = \text{clip}(1 - 0.05\,\delta_d \cdot h,\ 0.75,\ 1.0) \cdot Q_{\text{raw}},$$

where C is the comprehensive rubric score, M is the manual weighted score, h is the suspected hallucination rate, and δ_d is the domain-specific hallucination multiplier.

For reporting and visualization, we follow the same selection logic used in the dashboard. If the stopping metadata contains a `final_selection: best_of_last_k` record, we use that selected iteration when summarizing iteration-level behavior; otherwise, we fall back to the last available iteration. Thus, the signal values reported below correspond to the selected summary state rather than necessarily the final iteration that was executed.

In the reported runs, summarization, judging, and refinement all use `gpt-5-chat-latest`. Final evaluation aggregates repeated judge calls for stability, while the iterative refinement controller uses the single-run judge path inside the loop.

3.3 Results

Table3 reports per-lecture results for the main reference-free pipeline using dashboard-aligned selected-iteration reporting. The final score column gives the saved risk-adjusted score Q_{risk}, while the signal columns (h, coverage, and recall) are taken from the selected iteration used by the dashboard logic. We therefore report both the selected iteration and the total number of iterations completed.

Table: Main pipeline results across 7 lectures using dashboard-aligned selected-iteration reporting. Q_{risk} is the saved final risk-adjusted score in $[0, 1]$. Signals are taken from the selected iteration used by the dashboard logic.

Lecture	Domain	Q_{risk}	Sel. Iter	Ran	Stop	h	Cov	Recall
Lecture 1	Business	0.827	2	4	stalled	0.125	0.835	0.577
Lecture 2	Business	0.835	3	5	pass	0.167	0.983	0.667
Lecture 3	Humanities	0.837	7	7	pass	0.571	0.818	0.569
Lecture 4	Natural Sci.	0.858	2	4	pass	0.111	0.818	0.667
Lecture 5	Social Sci.	0.846	3	4	stalled	0.526	1.000	0.600
Lecture 6	Engineering	0.837	3	4	pass	0.722	0.970	0.613
Lecture 7	Natural Sci.	0.849	4	5	stalled	0.500	0.870	0.375
Average		**0.841**	**3.4**	**4.7**		**0.389**	**0.899**	**0.581**

The pipeline achieves a mean final score of 0.841 across all seven lectures under the reference-free tuned scoring policy. Under dashboard-aligned reporting, the average selected iteration is 3.4, while the average number of executed iterations is 4.7. This indicates that the final saved score is often associated with a summary state chosen from the best recent iterations rather than the literal last rewrite produced by the controller.

Scores remain tightly clustered in the range $[0.827, 0.858]$, suggesting that the tuned pipeline produces consistently strong outputs across diverse lecture domains. At the same time, variation in stopping state, hallucination rate, and glossary recall indicates that the retained summaries are not identical in grounding behavior even when their final scores remain relatively close.

3.3.1 Rubric Scores

Final rubric scores generally fall in the 4–5 range across all five dimensions, indicating that the judge consistently views the retained summaries as strong in overall structure, coverage, and readability. This narrow range suggests that the rubric layer is useful for confirming a strong quality baseline, but provides less discrimination among already-good outputs than the deterministic grounding signals do.

This pattern is consistent with the pipeline design: the final risk-adjusted score preserves rubric quality as a major component, while still allowing deterministic signals and hallucination-sensitive damping to separate lectures whose summaries differ more meaningfully in grounding behavior.

3.3.2 Hallucination Rate vs. Final Score

The selected-iteration hallucination rate varies substantially across lectures, from 0.111 to 0.722, but under the tuned scoring policy it does not dominate the final score as harshly as

in earlier penalty formulations. Instead, hallucination contributes through capped domain-aware damping, which preserves sensitivity to grounding risk without collapsing scores for summaries that rely on paraphrase or abstraction.

This is visible in lectures such as Lecture 6, which has the highest selected-iteration hallucination rate ($h = 0.722$) yet still achieves a strong final score of 0.837. Lecture 3 shows a similar effect: despite a relatively high selected-iteration hallucination rate ($h = 0.571$), its final score remains 0.837. These results suggest that the tuned policy penalizes weak grounding, but in a more controlled way than older harsher damping schemes.

3.3.3 Stopping Behavior

Stopping behavior varies across lectures even though final scores remain relatively close. In the runs shown here, four lectures terminate in a `pass` state, while three terminate as `stalled`. This indicates that the stopping controller is able to end refinement early when quality and signal conditions are met, but still identifies lectures where additional iterations are no longer yielding meaningful gains.

The distinction between *selected iteration* and *executed iterations* is also important. Several lectures use a best-of-last-k safeguard, meaning that the summary state ultimately retained for reporting is not always the final generated rewrite. This behavior is especially visible in Lectures 1, 2, 4, 5, 6, and 7, where the selected iteration precedes the final completed iteration.

3.3.4 Signal Coverage and Recall

Selected-iteration section coverage is high across the seven lectures, averaging 0.899, which indicates that the retained summaries usually cover most major lecture sections. Glossary recall is lower on average, at 0.581, suggesting that summaries more reliably preserve section-level topical breadth than full domain-specific terminology.

This tradeoff is particularly visible in Lecture 7, which retains strong section coverage (0.870) but has the lowest glossary recall (0.375). In that case, the selected summary appears to preserve broad structural coverage while sacrificing some domain-specific vocabulary. By contrast, Lecture 5 achieves perfect selected-iteration section coverage (1.000), showing that the refinement pipeline can preserve complete section-level breadth in some lectures even when other grounding signals remain imperfect.

4 Discussion

4.1 Grounding Signals Matter, but the Tuned Policy Reduces Over-Penalization

The most consistent pattern across the seven lectures is that deterministic grounding signals still provide more variance than the rubric layer, but under the tuned scoring policy they no longer dominate the final score as harshly as in earlier formulations. Final rubric scores remain concentrated in the 4–5 range across all five dimensions, which means that much of the remaining differentiation comes from signal-sensitive components such as suspected hallucination rate, section coverage, and glossary recall.

However, the current tuned policy applies a milder and capped domain-aware damping factor rather than the stronger penalty used in earlier versions of the pipeline. As a result, lectures with relatively high hallucination rates can still achieve strong final scores when the rubric and coverage signals remain strong. For example, Lecture 6 has the highest selected-iteration hallucination rate ($h = 0.722$) yet still achieves a final risk-adjusted score of 0.837, while Lecture 3 has $h = 0.571$ and also finishes at 0.837. This suggests that the current scoring formulation preserves sensitivity to grounding concerns without allowing the hallucination proxy to collapse otherwise strong summaries.

4.2 The Hallucination Proxy Remains Domain-Sensitive

Although the tuned policy reduces the severity of the penalty, the suspected hallucination signal still appears to behave differently across content types. Lecture 6 (engineering / SQL subqueries) is the clearest example: it has the highest selected-iteration hallucination rate in the cohort, even though its rubric and coverage signals remain strong. This likely reflects a limitation of the lexical grounding heuristic rather than a clear failure of summary quality. Technical lectures often require natural-language explanation of symbolic or structured source material, and such paraphrase can reduce lexical overlap even when the summary remains faithful.

A related effect appears in Lecture 3 (humanities), where the summary still scores well overall despite a relatively high hallucination rate. Abstract or argumentative lecture content often involves synthesis and paraphrase, which again weakens direct lexical overlap with the slides. In contrast, business-oriented lectures tend to use more list-based and definitional phrasing, making lexical alignment easier and resulting in lower hallucination rates. Taken together, these cases suggest that the suspected hallucination proxy should be interpreted as a *grounding-risk heuristic*, not as a direct factuality detector. More robust faithfulness checks, such as claim-level verification or natural language inference against slide content, would better separate stylistic paraphrase from genuine unsupported content.

4.3 The Rubric Layer Shows a Mild Ceiling Effect

Final rubric scores are consistently strong across all seven lectures, which indicates that the judge views the refined summaries as broadly well-structured, readable, and faithful. This is encouraging for overall system quality, but it also implies that the rubric layer has limited discriminative power among already-good summaries. In practice, once a summary reaches the 4–5 range across most dimensions, the rubric contributes less to cross-lecture separation than deterministic signals do.

This ceiling effect helps explain why final scores remain tightly clustered even when grounding-related signals vary more substantially. The rubric layer appears to function best as a confirmation that a summary meets a strong baseline of semantic and organizational quality, while the deterministic layer provides additional structure for distinguishing summaries that are similar in overall rubric quality but differ in lexical grounding behavior.

4.4 Adaptive Stopping Improves Efficiency, but Best-of-Last-k Matters

The stopping controller generally keeps the refinement process short. Under dashboard-aligned reporting, the average selected iteration is 3.4, while the average number of executed iterations is 4.7. This means the system usually reaches a strong candidate within a few rounds, even if the controller continues briefly before terminating. The difference between selected iteration and executed iterations also shows that the best-of-last-k safeguard is not merely cosmetic: in multiple lectures, the summary retained for reporting is not the last generated rewrite.

This behavior is important because it reduces end-of-run noise. A final rewrite can be slightly worse than a recent predecessor, especially when the refiner is making small local edits near convergence. By selecting from the most recent candidate pool rather than automatically returning the last rewrite, the pipeline gains robustness without requiring a more complex stopping mechanism.

4.5 Section Coverage and Glossary Recall Capture Different Notions of Completeness

The deterministic signals reveal a useful distinction between two forms of summary completeness. Selected-iteration section coverage is high across the seven lectures, averaging 0.899, which suggests that the retained summaries usually touch most major lecture sections. Glossary recall is lower, averaging 0.581, indicating that broad topical coverage is easier to preserve than full reproduction of domain-specific terminology.

Lecture 7 illustrates this tradeoff clearly: it maintains strong section coverage (0.870) but has the lowest glossary recall (0.375). In that case, the summary appears to preserve the main structure of the lecture while omitting some technical or methodological vocabulary. This is a reasonable tradeoff for student-facing summaries that prioritize broad comprehension, but it may be suboptimal in settings where technical terminology is itself a primary

learning objective. More generally, these results suggest that the current refiner tends to prioritize breadth of topic coverage over terminological precision when forced to remain concise.

4.6 Domain Routing Appears Plausible in the Current Runs

In the current saved outputs, the domain detector assigns lectures to five detected domains: business, humanities, natural sciences, social sciences, and engineering. These assignments are broadly plausible given the lecture content, and in particular Lecture 4 is routed to *natural sciences* rather than to a misaligned humanities rubric. This is an improvement over earlier concerns about coarse keyword-driven misclassification.

That said, domain routing remains a relatively brittle component because it relies on a fixed schema and lexical cues rather than richer semantic understanding. Interdisciplinary lectures could still be routed imperfectly, especially when the slide vocabulary mixes conceptual language with technical content. An LLM-based or hybrid classifier may ultimately provide more robust rubric routing for such cases.

4.7 Limitations

Small dataset. Seven lectures are not sufficient for strong statistical claims about domain-level behavior, stopping stability, or calibration of hallucination-sensitive scoring. The results should therefore be interpreted as descriptive evidence rather than as a conclusive benchmark.

Same model family for generation and evaluation. The system uses `gpt-5-chat-latest` for summarization, judging, and refinement. This creates a risk of style alignment between generator and evaluator, which may inflate rubric scores relative to what an independent model or human evaluator would assign.

No direct human evaluation in the main pipeline. Although the project retains optional reference-aware diagnostics, the main reported experiment is reference-free and does not include direct human ratings of summary usefulness. As a result, we cannot yet verify that the final score aligns closely with actual student preferences or expert pedagogical judgment.

Lexical grounding remains an imperfect proxy. The suspected hallucination rate is still a lexical heuristic. It can flag faithful but paraphrastic summaries as weakly grounded, especially in technical and argumentative lectures. It should therefore be interpreted as a useful risk signal rather than as a direct measure of fabrication.

Selected-state reporting complicates interpretation. Because the dashboard and analysis pipeline can report signals from the best selected iteration rather than the literal last executed iteration, final reporting mixes a retained-summary view with a controller-execution view. This is appropriate for evaluating the chosen summary, but it requires care when interpreting stopping dynamics and iteration counts together.

5 Conclusion

We presented an end-to-end pipeline for evaluating and improving LLM-generated lecture summaries in a default reference-free setting. The current system combines lever-based iterative refinement with adaptive stopping, domain-aware rubric evaluation, pairwise candidate selection, and a hybrid risk-adjusted scoring function that blends LLM judgment with deterministic grounding signals.

Across seven lecture slide decks spanning business, humanities, natural sciences, social sciences, and engineering, the pipeline achieves a mean final risk-adjusted score of 0.841 under the tuned scoring policy. Final summaries consistently receive strong rubric scores, while deterministic signals such as suspected hallucination rate, section coverage, and glossary recall provide additional structure for distinguishing summaries that are similar in overall rubric quality. The stopping controller also keeps refinement relatively efficient: under dashboard-aligned reporting, the average selected iteration is 3.4, while the average number of executed iterations is 4.7, showing that strong summary states are often reached within a small number of rounds.

Our results highlight two central tensions in reference-free lecture-summary evaluation. First, LLM-judged quality and lexical grounding signals are complementary but not interchangeable: summaries of abstract or technical lectures can receive strong rubric scores while still exhibiting elevated hallucination-risk signals because accurate paraphrase reduces direct lexical overlap with the slides. Second, the rubric layer exhibits a mild ceiling effect, which limits its ability to discriminate among already-strong outputs and motivates the use of hybrid scoring with deterministic diagnostics.

Several directions remain open for future work. Feeding suspected hallucination sentences directly back to the refiner as explicit correction targets could strengthen the connection between grounding diagnostics and revision behavior. More robust domain routing, potentially through an LLM-based or hybrid classifier, could improve rubric selection for interdisciplinary lectures. Expanding the lecture corpus would allow more reliable comparison across subject areas and refinement behaviors. Finally, validating the final score against human judgments would help establish whether the pipeline's notion of quality aligns with what students and instructors actually find useful in lecture summaries.

References

Breiman, Leo. 2001. "Statistical Modeling: The Two Cultures." *Statistical Science* 16(3): 199–215. [Link]

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. [Link]

Gallagher, Shannon, Swati Rallapalli, and Tyler Brooks. 2025. "Evaluating LLMs for Text Summarization: An Introduction." Online blog post, April 7. [Link]

Hein, David, AlanaLee Christie, Michael Holcomb, Bingqing Xie, A.J. Jain, Joseph Vento, Neil Rakheja, AmeerHamza Shakur, Scott Christley, LindsayG. Cowell, James Brugarolas, AndrewR. Jamieson, and Payal Kapur. 2025. "Iterative refinement and goal articulation to optimize large language models for clinical information extraction." *npj Digital Medicine* 8(1), p. 301. [Link]

Karras, Tero, Samuli Laine, and Timo Aila. 2019. "A Style-Based Generator Architecture for Generative Adversarial Networks." In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [Link]

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, AidanN. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Advances in Neural Information Processing Systems*. [Link]

Yang, Yujing, Boqi Chen, Kua Chen, Gunter Mussbacher, and Daniel Varró. 2024. "Multi-step Iterative Automated Domain Modeling with Large Language Models." In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS) Companion*. Association for Computing Machinery. [Link]

Zhang, Yutong, Lixing Chen, Shengdong Li, Nan Cao, Jiaxing Ding, Zhe Qu, and Yang Bai. 2024. "Way to Specialist: Closing Loop Between Specialized LLM and Evolving Domain Knowledge Graph." *arXiv preprint* abs/2411.19064. [Link]