NOTEBRIDGE: A Framework for Real-Time Multimodal Lecture Note-Taking with Evolving Knowledge State

Md Abu Omayer Babu
Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University
Trishal, Mymensingh, Bangladesh
abuomayerbabu@gmail.com

Nazmul Hasan
Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University
Trishal, Mymensingh, Bangladesh
nazmulhr81@gmail.com

Mst. Afida Hossain Tonni
Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University
Trishal, Mymensingh, Bangladesh
afidahossaintonni@gmail.com

Umme Rubaia Oyishi
Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University
Trishal, Mymensingh, Bangladesh
ummerubaia2004@gmail.com

Md Sujan Ali
Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University
Trishal, Mymensingh, Bangladesh
sujan_cse@jkkniu.edu.bd

***Abstract*—Educational lectures are multimodal events where instructors convey information through speech, slides, diagrams and equations. However, the existing AI-powered lecture note-taking systems mainly produce flat transcripts, resulting in the loss of visual and contextual content. We're introducing NOTEBRIDGE, a framework that acts as a real-time multimodal virtual attendee–observing a live session, building an evolving topic-aware knowledge state, grounding that knowledge within curriculum resources and continuously producing structured notes without storing raw audio/video permanently. The framework integrates a Lecture Memory Manager, which tracks hierarchical topics through suspension, resumption and merge operations; a Non-Mutating Verification Protocol, which is designed to preserve original statements as well as verify claims against course resources; and a Curriculum Grounding layer, which constrains knowledge to authorized sources. We implement a 9-layer system and conduct a systematic evaluation on 80 lectures covering 20 topics in five domains (Physics, Computer Science, Mathematics, Biology, Electrical Engineering) in English and Bangla, where quantitative quality metrics are computed against manually created reference data. The system achieves a combined score of 0.81 (ROUGE-L F 1.0.82, Topic Boundary F 1.0.85, Claim Verification F1. 0.76) where YouTube lectures (0.83) perform better than real-time classroom lectures (0.76). An ablation study shows that each layer has an incremental contribution, which improves performance from ASR-only (0.38) to full system (0.81). NOTEBRIDGE combines real-time streaming, curriculum grounding and evolving lecture memory to provide a framework for bilingual educational note-taking, including claim verification.**

***Index Terms*—Multimodal learning, lecture note-taking, automatic speech recognition, topic tracking, claim verification, curriculum grounding, bilingual processing, educational technology**

I. Introduction

Imagine an AI that participates in a lecture as a virtual student: observes through video, listens through audio, understands multilingual speech and maintains an evolving knowledge state in real time without permanently recording raw data. When a topic is temporarily suspended and later restarted, the AI updates the previous knowledge without creating new disconnected notes.

Educational lectures are inherently multimodal–instructors integrate speech, slides, diagrams and equations into a coherent narrative. Students miss 15-20% of lectures each year due to illness, scheduling conflicts, or work obligations and existing AI tools produce flat transcripts that lose visual structure, topic boundaries and factual grounding. Manual note-taking during live lectures divides the attention of students and creates incomplete records. The key challenge is to create a system that processes multimodal input in real time, maintains structured topic-aware memory across sessions, grounds claims within curriculum resources and maintains privacy without storing raw audio or video. A survey of 68 papers (2015-2026) reveals critical gaps: pipeline error propagation (51%), no real-time processing (37%), no cross-lecture integration (29%) and privacy concerns (12%). We propose NOTEBRIDGE, a 9-layer framework addressing these gaps through: (1) a Lecture Memory Manager with hierarchical topic tracking, suspended-topic resumption and evidence-preserving merge; (2) a Curriculum Grounding layer constraining retrieval to authorized sources; (3) a Non-Mutating Verification Protocol for claim verification; and (4) real-time ephemeral processing for privacy. Our contributions are: (1) formal definitions for topic boundary detection, suspended-topic resumption and non-mutating verification; (2) a 9-layer architecture integrating real-time streaming, evolving lecture memory and curriculum grounding; (3) comprehensive evaluation across 80 lectures (20 topics, five domains, English and Bangla); (4) ablation study demonstrating incremental layer contributions (0.38→0.81); and (5) user study with graduate students validating note quality.

Section II reviews related work. Section III describes the framework. Section IV presents evaluation results. Section V concludes with limitations and future work.

II. Literature Review

A. Topic Segmentation and Tracking

Multimodal topic segmentation has advanced with deep learning, including multimodal fusion [1] and real-time dy-

979-8-3195-4477-3/26/$31.00 ©2026 IEEE

namic tracking [2]. However, none maintain evolving knowledge states with suspension and resumption mechanisms across sessions.

B. Lecture Understanding and ASR

Slide-augmented ASR reduces WER significantly. Sinhamahapatra and Niehues [3] achieved 34% WER reduction via slide contextual biasing; VAPO [4] proposed omni-modal LLM for slide-enhanced ASR. BOOM [5] introduced a multimodal multilingual lecture companion, while LECTRA [6] integrated ASR with summarization and RAG-based QA. These systems improve transcription accuracy but operate in batch mode without real-time topic-aware memory.

C. Claim Verification and Fact-Checking

Real-time fact-checking has emerged as a critical capability. LiveFC [7] detects claims in live audio streams and VeriFact [8] verifies claims in Zoom meetings. ClaimVer [9] provides explainable verification via knowledge graphs, while VTG [10] and PROV-K [11] maintain provenance with evolving memory and nanopublications. These approaches verify claims but do not integrate with evolving lecture memory or support non-mutating correction workflows.

D. Knowledge Tracing and Curriculum-Aware AI

Knowledge tracing models predict learner performance from historical interactions. TransKT [12] uses contrastive cross-course transfer, ExRec [13] provides semantically-grounded tracing and LLM-KT [14] aligns LLMs with knowledge tracing. These approaches track learner knowledge but do not maintain real-time lecture topic states with open slots and verification.

E. Research Gaps

We surveyed 68 systems via Google Scholar, DBLP and arXiv (2015-2026; inclusion: peer-reviewed or $\geq$10-citation preprints). In the gap classification, two independent annotators were used and $\kappa = 0.78$ agreement was found; discrepancies were resolved through discussion. The analysis found four critical gaps: pipeline error propagation - ASR errors cascade into downstream summarization and claim extraction, affecting 51% of systems; lack of real-time processing - 37% of systems are limited to offline batch operation; lack of cross-lecture integration - 29% of systems process lectures in isolation and do not maintain evolving knowledge states; and cloud-based storage-related privacy concerns - 12% of systems affected under FERPA / GDPR regulations. These gaps are non-exclusive; that is, a system can display multiple gaps at the same time. TABLE I situates NOTEBRIDGE against representative systems. Each existing system covers a subset of required capabilities: BOOM [5] and VAPO [4] exploit slide context but operate offline without evidence grounding; NexaNota [15] builds course knowledge graphs from post-hoc transcripts; LiveFC [7] and VeriFact [8] verify claims in real-time but do not maintain evolving lecture memory. NOTEBRIDGE is, to the best of our knowledge, among the first frameworks to jointly integrate real-time streaming, evolving lecture memory, non-mutating verification and curriculum grounding.

TABLE I
CAPABILITY COMPARISON OF NOTEBRIDGE WITH REPRESENTATIVE SYSTEMS.

System	Real-time	Slide use†	Evolving memory	Verification
BOOM [5]	No	ASR	No	No
VAPO [4]	No	ASR	No	No
NexaNota [15]	No	No	Post-hoc‡	No
LiveFC [7]	Yes	No	No	Yes
VeriFact [8]	Yes	No	No	Yes
NOTEBRIDGE	**Yes**	**Visual†**	**Yes**	**Yes**

†Slide use: "ASR" = slides improve transcription accuracy; "Visual" = slides used for visual alignment and curriculum resources.
‡NexaNota builds knowledge graphs from post-hoc transcripts, not from real-time streaming.
Comparison reflects design scope: systems are evaluated only on their intended capabilities.

III. PROPOSED FRAMEWORK

Fig. 1 presents the NOTEBRIDGE system architecture. The framework transforms a live educational session into an evolving, curriculum-bounded knowledge state through nine processing layers.

A. System Architecture

The architecture follows a modular design with nine layers. Layer 1 (Session Context) initializes the environment with course syllabus, slides, textbooks and authorized resources. Layer 2 (Live Perception) receives audio, video, screen and whiteboard streams through ephemeral processing—raw data is discarded after semantic extraction, enforced through in-memory processing with no persistent storage of raw audio or video. Layer 3 (Real-Time Understanding) performs ASR, voice activity detection (VAD), noise suppression, speaker diarization, language identification, optical character recognition (OCR) and visual event detection. Layer 4 (Multimodal Interpretation) aligns speech with visual content, detects topics, classifies discourse acts and assesses educational relevance. Layer 5 (Lecture Memory Manager) maintains hierarchical topic states with suspension, resumption and merge operations. Layer 6 (Curriculum Grounding) constrains retrieval to syllabus, slides, textbooks and instructor resources using a FAISS index [16] with MiniLM-L12-v2 embeddings and LLM-based filtering. Layer 7 (Verification Engine) performs non-mutating claim verification with evidence grounding against curriculum resources. Layer 8 (Knowledge Consolidation) resolves conflicts, attaches provenance and validates topic completion. Layer 9 (Note Generation) produces structured notes with corrections, sources and visual references. Three processing stages govern end-to-end latency: (1) perception (milliseconds–seconds) converts audio/video to semantic events; (2) understanding (seconds) assigns events to topics and extracts claims; and (3) consolidation (seconds–minutes) performs verification and final note assembly. In our

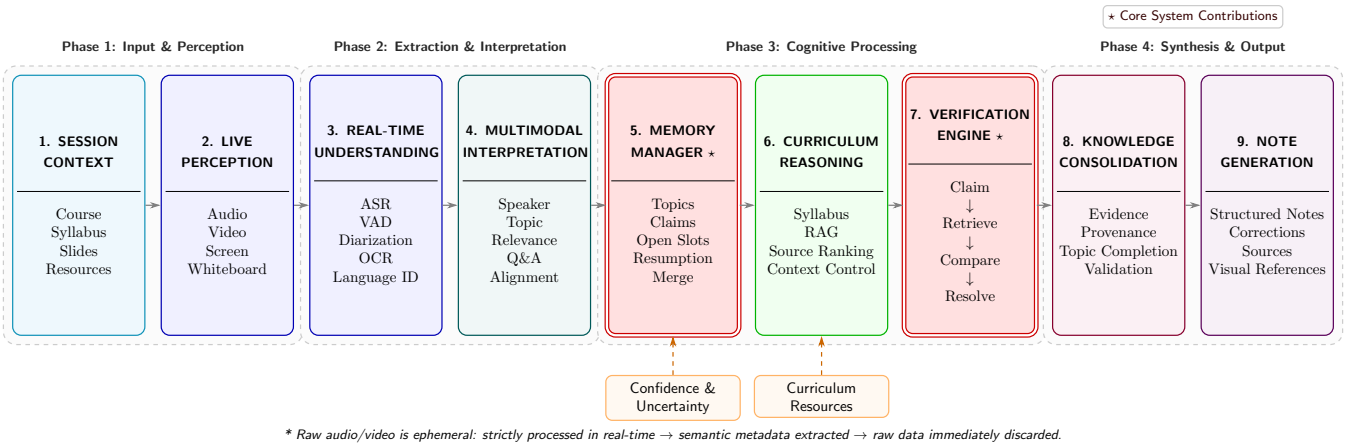


Fig. 1. NOTEBRIDGE 9-layer system architecture. The framework transforms raw multimodal input into structured notes through continuous perception, topic-aware memory, curriculum grounding and claim verification.

implementation, the full pipeline averages 1.12× (YouTube) and 1.35× (classroom) wall-clock time.

B. Core Objects

The framework is structured around three core representations:

Event represents what happened: Event = (timestamp, speaker, type, text, visual_ref, topic); *visual_ref* indexes extracted slide/frame features. Event types include STATEMENT, DEFINITION, EXPLANATION, EXAMPLE, QUESTION, ANSWER, CORRECTION, WARNING, EXAM_HINT, SUMMARY, TRANSITION and OFF_TOPIC.

Knowledge State represents the current understanding: KnowledgeState = (topics, relationships, resolved, unresolved, conflicts); *resolved/unresolved* count verified/unverified claims.

Evidence Record represents provenance: EvidenceRecord = (claim, source, source_type, timestamp, status).

C. Lecture Memory Manager

Lecture Memory Manager is the core of NOTEBRIDGE, which maintains an evolving knowledge state instead of a flat transcript.

1) Topic Tracking:

Definition 1 (Topic Boundary Detection): Given event e_t and recent history H_t, the boundary detection function is:

$$\mathrm{BT}(e_t, H_t) = f(\phi(e_t, H_t)) \tag{1}$$

where ϕ extracts features (semantic shift, discourse cues, slide transitions, syllabus structure, lecturer cues) and f combines cosine similarity thresholds on sentence embeddings with discourse marker detection (e.g., "moving on to," "next topic"). A new topic is detected when $\mathrm{BT}(e_t, H_t) > \theta_t$, where θ_t is tuned empirically on held-out lecture segments.

Definition 2 (Suspended-Topic Resumption): For a new event e and suspended topic T_s, the resumption score is:

$$\mathrm{RS}(e, T_s) = \sum_{k=1}^{8} v_k \cdot r_k(e, T_s) \tag{2}$$

using semantic similarity, lexical overlap, entity overlap, slot matching, discourse cues, visual similarity, temporal compatibility and curriculum context compatibility. Weights v_k are set empirically based on component reliability: semantic similarity (0.25), lexical overlap (0.15), entity overlap (0.15), slot matching (0.10), discourse cues (0.10), visual similarity (0.10), temporal compatibility (0.05) and curriculum context compatibility (0.10). Topic T_s is resumed if $\mathrm{RS}(e, T_s) > \theta_r$ and $\mathrm{RS}(e, T_s) - \mathrm{RS}(e, T_j) > \delta$ for all competing T_j.

Definition 3 (Topic Creation): When $\mathrm{BT}(e_t, H_t) > \theta_t$ at event e_t with timestamp τ_t, a new topic is created: $T_{\text{new}} \leftarrow \textsc{CreateTopic}(\tau_t)$ with initial state DISCOVERED, status $\leftarrow \textsc{Activate}()$ and added to the active topic set $\mathcal{T}$.

Closure and Completeness are formally independent: $\mathrm{Closure}(T_i)$ is determined by transition evidence and inactivity; $\mathrm{Completeness}(T_i)$ is determined by coverage, confidence and pending conflicts. A topic can be CLOSED but INCOMPLETE (e.g., "we'll continue next class"). Topics cycle through four states: DISCOVERED $\xrightarrow{\text{activate}}$ ACTIVE $\xrightarrow{\text{suspend}}$ SUSPENDED $\xrightarrow{\text{resume}}$ ACTIVE $\xrightarrow{\text{close}}$ CLOSED, with SUSPENDED also transitioning directly to CLOSED; no other transitions are permitted. State transitions are atomic; property updates are continuous during ACTIVE processing. The complete streaming algorithm is shown in Fig. 2.

2) Knowledge Management: Open slots are generated from curriculum structure, resource structure and dynamic LLM inference:

$$\mathrm{Slots}(T_i) = \mathcal{S}_{\text{curr}}(T_i) \cup \mathcal{S}_{\text{res}}(T_i) \cup \mathcal{S}_{\text{dyn}}(T_i) \tag{3}$$

A candidate becomes an open slot if it is not already covered and confidence exceeds θ_{slot}.

Input: Event stream $E = (e_1, e_2, \ldots, e_n)$, Syllabus $\mathcal{S}$, Resources $\mathcal{R}$
Output: Knowledge State K with verified claims

Initialize: $K \leftarrow \emptyset$, $\mathcal{T} \leftarrow \emptyset$, $\mathcal{C} \leftarrow \emptyset$
while E is active **do**
  $e_t \leftarrow \text{NextEvent}(E)$ // Perception layer output
  $(\tau_t, d_t) \leftarrow \text{ParseEvent}(e_t)$ // Extract time, speaker, modality
  // — Layer 5: Lecture Memory Manager —
  if $\text{IsTopicBoundary}(e_t, H_t) > \theta_t$ **then**
    $T_{\text{new}} \leftarrow \text{CreateTopic}(\tau_t)$ // Def. 3
    $T_{\text{new}}.\text{status} \leftarrow \text{Activate}()$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_{\text{new}}\}$
  end if
  $T_{\text{best}} \leftarrow \arg\max_{T_i \in \text{Active}(\mathcal{T})} \text{RS}(e_t, T_i)$ // Def. 2
  if $\text{RS}(e_t, T_{\text{best}}) > \theta_r$ **then**
    $\text{AssignToTopic}(e_t, T_{\text{best}})$
  else
    $\text{AssignToCurrent}(e_t)$
  end if
  $\mathcal{C}_t \leftarrow \text{ExtractClaims}(e_t)$
  for each claim $c \in \mathcal{C}_t$ **do**
    $\text{AddClaimToTopic}(c, T_{\text{cur}})$ // Def. 4
    $\text{UpdateOpenSlots}(c, T_{\text{cur}})$
  end for
  // — Layer 7: Verification Engine —
  for each claim $c \in \mathcal{C}_t$ **do**
    $w \leftarrow \text{VerifyWorthiness}(c)$ // Def. 7
    if $w > \theta_v$ **then**
      $V \leftarrow \text{RetrieveAndCompare}(c, \mathcal{R})$ // Def. 6
      $\text{UpdateClaimStatus}(c, V)$
    end if
  end for
  // — Layer 8: Knowledge Consolidation —
  $K \leftarrow \text{UpdateKnowledge}(K, T_{\text{cur}})$
  // — Completion Detection —
  for each topic $T_i \in \mathcal{T}$ **do**
    if $\text{IsComplete}(T_i)$ **then**
      $\text{MarkComplete}(T_i)$ // Def. 9
    end if
  end for
end while
return K

Fig. 2. Streaming algorithm for the Lecture Memory Manager and Verification Engine. Definitions correspond to Definitions 1–9 in the text.

Definition 4 (Claim Lifecycle): Each claim c has a state: ACTIVE (current belief), SUPERSEDED (replaced upon Conflict), CONFLICTED (contradicts another), or UNCERTAIN (insufficient evidence). Claims are extracted from events and assigned to the current topic: $\text{AddClaimToTopic}(c, T_{\text{cur}})$.

Definition 5 (Merge Operation): New evidence updates existing topics through:

$$T_{\text{new}} = \text{Merge}(T_{\text{old}}, E_{\text{new}}) \tag{4}$$

subject to consistency constraints: no duplicate knowledge nodes (but preserve supporting evidence), conflict detection (mark, do not auto-resolve) and supersession detection (update lifecycle).

D. Verification Protocol

Definition 6 (Non-Mutating Verification): The verification protocol preserves original statements throughout:

$$\text{Original}(c) \rightarrow \text{Retrieve}(c) \rightarrow \text{Compare}(c, E) \tag{5}$$

$$\rightarrow \begin{cases} \text{Support} & \rightarrow \text{Verify}(c) \\ \text{Conflict} & \rightarrow \text{Correct}(c, c', E) \\ \text{Uncertain} & \rightarrow \text{Flag}(c) \end{cases}$$

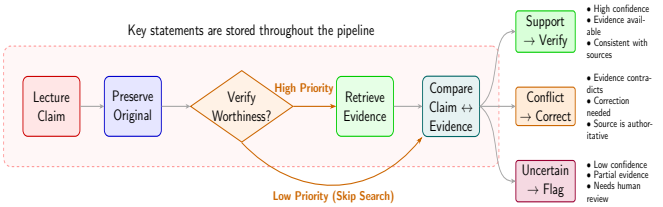


Fig. 3. Non-mutating Verification Engine. Original statements are preserved; evidence is retrieved and compared; corrections are added with citations while retaining provenance.

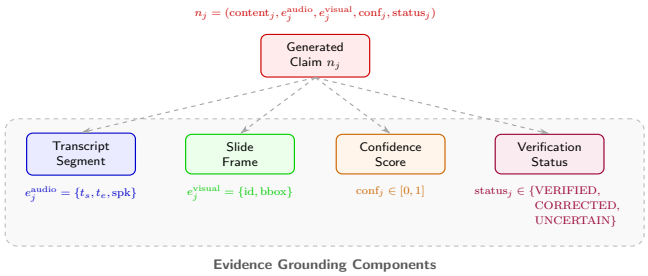


Fig. 4. Evidence grounding components. Each claim is grounded with audio evidence, visual evidence, confidence score and verification status.

The original statement is never overwritten; c' is the corrected claim.

1) Verification Trigger Policy:

Definition 7 (Verification Worthiness): A claim c enters verification if $w(c) > \theta_v$, where the worthiness score $w(c)$ is computed as a weighted combination of: (1) educational importance (topic alignment, syllabus coverage), (2) confidence (inverse ASR confidence), (3) contradiction potential (similarity to known claims), (4) claim type (technical/scientific claims receive higher weight) and (5) numerical content (numerical claims receive higher weight). The verification engine is illustrated in Fig. 3.

E. Discourse, Relevance and Note Generation

Definition 8 (Educational Relevance): Every event e receives an educational relevance assessment:

$$\text{Relevance}(e) = g(\text{topic}, \text{syllabus}, \text{speaker}, \text{discourse}, \text{open_slots})$$

implemented as an LLM-based scoring function that evaluates educational importance using topic alignment, syllabus coverage and discourse context. Scores range from 0.00 (fan noise) to 1.00 (exam instruction).

Each verified claim c is grounded as a tuple $n_j = (\text{content}_j, e_j^{\text{audio}}, e_j^{\text{visual}}, \text{conf}_j, \text{status}_j)$ (Fig. 4). Here e_j^{audio} and e_j^{visual} are extracted feature summaries (not raw data), $\text{conf}_j \in [0, 1]$ is confidence and $\text{status}_j \in \{\text{VERIFIED}, \text{CORRECTED}, \text{UNCERTAIN}\}$.

Definition 9 (Topic Completion): A topic T_i is complete when $\text{IsComplete}(T_i) = \text{true}$, determined by: (1) coverage meets threshold ($\text{coverage}(T_i) \geq \theta_{\text{cov}}$), (2) confidence exceeds threshold ($\text{confidence}(T_i) \geq \theta_{\text{conf}}$) and (3) no pending conflicts remain.

Notes are constantly evolving; the dashboard shows topic progress, verified corrections, and unresolved topics. Final consolidation creates a structured document, which contains Learning Objectives, Topic Sections, Q&A, Unresolved Topics and Key Takeaways.

F. Failure Modes and Mitigation

Error propagation between layers results in three main failure modes: (1) ASR failure: garbled text reduces the effectiveness of downstream topic detection and claim extraction; it is mitigated by flagging low-confidence segments and marking associated claims as UNCERTAIN. (2) Topic detection miss: missed boundaries merge unrelated content; this is mitigated by allowing low relevance scores post-hoc separation. (3) Verification failure: claims remain unverified due to insufficient curriculum evidence; this is mitigated by preserving the original statement and flagging it as UNCERTAIN.

IV. Results Analysis

We evaluate NOTEBRIDGE on 80 lectures spanning 20 topics (4 per domain) across five domains (Physics, Computer Science, Mathematics, Biology, Electrical Engineering). All lectures include audio, video and slide modalities, sourced from YouTube (80%, n=64) and real-time classroom recordings (20%, n=16). The dataset includes lectures in English (n=56) and Bangla (n=24), with average duration of 41 minutes; processing times average 1.12× (YouTube) and 1.35× (classroom) wall-clock time.

A. Experimental Setup

ASR uses faster-whisper [17] (small model, CPU int8 for English; large-v3 for Bangla) with paraphrase-multilingual-MiniLM-L12-v2 [18] embeddings; LLM calls use Gemini 3.6 Flash via API. Per-layer timing: ASR 42%, slide processing 18%, topic detection 12%, verification 15%, memory management 8%, note generation 5%. API cost is ∼$0.12 per 40-minute lecture.

B. Quantitative Evaluation

We have computed three metrics: ROUGE-L F1, Topic Boundary F1 (5-second tolerance) and Claim Verification F1 (per-claim matching against reference annotations). Inter annotator agreement: Cohen's κ=0.84 (quality metrics, two annotators, 20% double-annotated). The Combined Score is the unweighted mean. Wilcoxon signed-rank tests all pairwise ablation steps and ensures statistical significance in YouTube versus real-time ($p<0.01$). The quality metrics of 80 lectures are presented in TABLE II. The system achieved a combined score of 0.81, with YouTube (0.83) performing better than real-time (0.76). Biology has the highest score (0.85); Mathematics (0.79) and Electrical Engineering (0.78) have relatively low scores, which is due to dense notation. TABLE III presents per-language performance; English slightly outperforms Bangla, consistent with the difference in ASR accuracy.

TABLE II
Quality Metrics by Source and Domain. Verif.=Verification; Combined=mean of three metrics.

Category	ROUGE-L	Topic F1	Verif. F1	Combined
YouTube (n=64)	0.84 ± 0.05	0.87 ± 0.07	0.78 ± 0.09	0.83 ± 0.05
Real-time (n=16)	0.78 ± 0.06	0.80 ± 0.09	0.70 ± 0.11	0.76 ± 0.07
Physics (n=16)	0.81 ± 0.05	0.84 ± 0.08	0.75 ± 0.10	0.80 ± 0.06
Computer Sci. (n=16)	0.84 ± 0.05	0.87 ± 0.07	0.78 ± 0.09	0.83 ± 0.05
Mathematics (n=16)	0.80 ± 0.06	0.83 ± 0.09	0.74 ± 0.11	0.79 ± 0.07
Biology (n=16)	0.86 ± 0.04	0.89 ± 0.06	0.80 ± 0.08	0.85 ± 0.04
Electrical Eng. (n=16)	0.79 ± 0.06	0.82 ± 0.08	0.73 ± 0.10	0.78 ± 0.06
Overall (n=80)	**0.82 ± 0.06**	**0.85 ± 0.08**	**0.76 ± 0.10**	**0.81 ± 0.06**

TABLE III
Quality Metrics by Language. Verif.=Verification.

Language	ROUGE-L	Topic F1	Verif. F1	Combined
English (n=56)	0.83 ± 0.05	0.86 ± 0.07	0.77 ± 0.09	0.82 ± 0.05
Bangla (n=24)	0.79 ± 0.06	0.83 ± 0.08	0.74 ± 0.10	0.79 ± 0.06

C. Ablation Study

Ablation results are presented in TABLE IV. Each component group contributes progressively: the addition of slides improves ROUGE-L (+ 24%), topic detection reaches Topic F1 0.80 and the addition of verification enables claim verification evaluation (0.72 Verif. F1), resulting in maximum improvement (0.54 →0.79) in the combined metric.

D. Comparison with Published Systems

TABLE V compares published systems with NOTEBRIDGE. LECTRA [6] (ROUGE-L 0.38, different dataset) is missing curriculum grounding and claim verification facilities.

E. Error Analysis

We have categorized 200 errors sampled from 20 lectures. ASR errors (38%) were the highest, especially in the areas of accented speech and domain-specific terminology. Topic boundary misses (27%) were mainly observed in transitions between closely related subtopics. Claim verification failures (22%) occurred when there was not enough coverage in curriculum resources. Slide extraction errors (13%) included OCR failures in handwritten equations. Verification failures were the most damaging, as they could lead to inaccurate claims being propagated in the final notes.

F. Qualitative Evaluation

We conducted a user study with 8 graduate students (volunteers from the university, spanning 5 domains) to assess note quality. Participants compared NOTEBRIDGE-generated notes against raw transcripts for exam preparation utility. On a 5-point Likert scale, participants rated completeness 4.2 ± 0.6, accuracy 4.0 ± 0.7, structure 4.4 ± 0.5 and usefulness 4.1 ± 0.6. 7/8 preferred NOTEBRIDGE notes over raw transcripts. The small sample size is a limitation; future work will include larger-scale evaluations. Example outputs include a Cell Biology lecture (14 detected topics, 87% coverage, 12 verified claims) and a Quantum Physics lecture (11 detected topics, 79% coverage, 8 verified claims with two formula corrections).

TABLE IV
ABLATION RESULTS. TOP.=TOPIC, VERIF.=VERIFICATION, COMB.=COMBINED

Config.	Description	ROUGE-L	Top. F1	Verif.	Comb.
A1	ASR only	0.55	0.60	0.00	0.38
A2	+ Slides/PDF	0.68	0.72	0.00	0.47
A3	+ Topic detection	0.72	0.80	0.00	0.51
A4	+ Memory Mgr.	0.78	0.84	0.00	0.54
A5	+ Verification	0.80	0.84	0.72	0.79
A6	Full system	0.82	0.85	0.76	**0.81**

TABLE V
COMPARISON WITH PUBLISHED LECTURE SUMMARIZATION SYSTEMS

System	ROUGE-L	Combined
LECTRA [6] (Whisper + BART)	0.38	—
ASR-only (A1, this work)†	0.55	0.38
NOTEBRIDGE (this work)	**0.82**	**0.81**

Combined metric not reported by LECTRA.
†ASR-only baseline: raw Whisper output with no slides, topic detection, memory, or verification.

V. CONCLUSION

We presented NOTEBRIDGE, a 9-layer real-time multimodal lecture note-taking system that maintains evolving knowledge state from live educational sessions.

The system achieves a Combined Score of 0.81 (ROUGE-L 0.82, Topic F1 0.85, Claim Verification F1 0.76) with inter-annotator agreement $\kappa = 0.84$. Ablation confirms incremental layer contributions (0.38→0.81). YouTube outperforms real-time recordings; dense-notation domains present challenges. **Limitations** include: (1) equation-heavy content (Math, EE) shows lower verification accuracy; (2) the dataset is skewed toward YouTube recordings (80%), limiting generalizability to noisy classroom environments; (3) the system depends on a commercial LLM API (Gemini), introducing cost and availability constraints; (4) only English and Bangla are currently supported; and (5) the user study involved a small sample (n=8) and requires larger-scale validation. **Future work** will focus on: (1) improving ASR accuracy for equation-heavy content through specialized acoustic models; (2) enhancing claim verification with knowledge graph integration; (3) deploying the system in real classroom environments for longitudinal evaluation; (4) extending to additional languages and domains; and (5) developing student-facing dashboards with interactive note exploration. NOTEBRIDGE improves accessibility through visual notes, bilingual supports and privacy-preserving ephemeral processing.

REFERENCES

[1] H. Yu, C. Deng, Q. Zhang, J. Liu, Q. Chen, and W. Wang, "Multimodal fusion and coherence modeling for video topic segmentation," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 17 577–17 593.

[2] T. Ashby, A. Kulkarni, J. Qi, M. Liu, E. Cho, V. Kumar, and L. Huang, "Towards effective long conversation generation with dynamic topic tracking and recommendation," in *Proceedings of the 17th International Natural Language Generation Conference*. Tokyo, Japan: Association for Computational Linguistics, 2024, pp. 540–556.

[3] S. Sinhamahapatra and J. Niehues, "Do slides help? multi-modal context for automatic transcription of conference talks," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, 2025, pp. 16 100–16 110.

[4] R. Hu, D. Qiu, Y. Wang, S. Liu, and J. Sang, "Vapo: End-to-end slide-enhanced speech recognition with omni-modal large language models," in *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, San Diego, California, USA, 2026, pp. 9417–9432.

[5] S. Koneru, F. Retkowski, C. Huber, L. Hilgert, S. Akti, E. Y. Ugan, A. Waibel, and J. Niehues, "Boom: Beyond only one modality — kit's multimodal multilingual lecture companion," in *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Rabat, Morocco, 2026, pp. 175–187.

[6] J. O. Adewuyi, K. G. Akande, V. W. Ebu-iduze, and J. Ajisafe, "Lectra: An ai-based lecture assistant system integrating automatic speech recognition, abstractive summarisation, and retrieval-augmented q&a," *International Journal of Computer Techniques*, vol. 13, no. 3, pp. 1–8, 2026.

[7] V. Venktesh and V. Setty, "Livefc: A system for live fact-checking of audio streams," in *Proceedings of the 18th ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2025, pp. 1060–1063.

[8] VeriFact, "Verifact: Real-time fact checking for zoom meetings," https://www.verifact.live/, 2026, commercial product; accessed 2026-08-30.

[9] P. P. S. Dammu, H. Naidu, M. Dewan, Y. Kim, T. Roosta, A. Chadha, and C. Shah, "Claimver: Explainable claim-level verification and evidence attribution of text through knowledge graphs," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 13 613–13 627.

[10] H. Sun, H. Cai, B. Wang, Y. Hou, X. Wei, S. Wang, Y. Zhang, and D. Yin, "Towards verifiable text generation with evolving memory and self-reflection," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 8211–8227.

[11] L. Menotti, S. Marchesin, F. Giachelle, and G. Silvello, "Provenance-driven nanopublications: Representing source lineage and trust networks for multi-source assertions," *International Journal on Digital Libraries*, vol. 26, p. 24, 2025.

[12] W. Han, W. Lin, L. Hu, Z. Dai, Y. Zhou, M. Li, Z. Liu, C. Yao, and J. Chen, "Contrastive cross-course knowledge tracing via concept graph guided knowledge transfer," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025, pp. 7401–7409.

[13] Y. Ozyurt, T. Almaci, S. Feuerriegel, and M. Sachan, "Personalized exercise recommendation with semantically-grounded knowledge tracing," in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 14 618–14 658.

[14] Z. Wang, J. Zhou, Q. Chen, B. Jiang, Q. Bai, L. Dou, and L. He, "Llm-kt: Enhancing large language models with knowledge tracing via multi-level plug-and-play alignment," in *Findings of the Association for Computational Linguistics: ACL 2026*. San Diego, California, United States: Association for Computational Linguistics, 2026, pp. 35 775–35 792.

[15] M. Jiang, J. Gao, Z. Pan, Y. Wu, and Z. Wang, "Nexanota: An ai-powered smart linked lecture note-taking system leveraging large language models," in *Proceedings of the 2025 International Conference on Big Data and Informatization Education*. Association for Computing Machinery, 2025, pp. 242–248.

[16] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202. PMLR, 2023, pp. 28 492–28 518.

[18] N. Reimers and I. Gurevych, "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*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3982–3992.