Article

Not peer-reviewed version

STREAM: A Semantic Transformation and Real-Time Educational Adaptation Multimodal Framework in Personalized Virtual Classrooms

Leyli Nouraei Yeganeh *, Yu Chen *, Nicole Scarlett Fenty, Amber Simpson, Mohsen Hatami

Posted Date: 27 October 2025

doi: 10.20944/preprints202510.2065.v1

Keywords: multimodal learning; adaptive learning; personalized learning; virtual classrooms; speech recognition; transformer-based NLP; computer vision; Universal Design for Learning; learner modeling

 doi:10.20944/preprints202510.2065.v1

Article

STREAM: A Semantic Transformation and Real-Time Educational Adaptation Multimodal Framework in Personalized Virtual Classrooms

Leyli Nouraei Yeganeh [1,*], Yu Chen [2*], Nicole Scarlett Fenty [1], Amber Simpson [1] and Mohsen Hatami [2]

[1] Department of Teaching, Learning and Educational Leadership, Binghamton University, Binghamton, NY
[2] Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY
* Correspondence: lnourae1@binghamton.edu (L.N.Y.); ychen@binghamton.edu (Y.C.)

Abstract: Most adaptive learning systems personalize around content sequencing and difficulty adjustment rather than transforming instructional material within the lesson itself. This paper presents the STREAM (Semantic Transformation and Real-Time Educational Adaptation Multimodal) framework. This modular pipeline decomposes multimodal educational content into semantically tagged, pedagogically annotated units for regeneration into alternative formats while preserving source traceability. STREAM integrates automatic speech recognition, transformer-based natural language processing, and planned computer vision components to extract instructional elements from teacher explanations, slides, and embedded media. Each unit receives metadata including timecodes, instructional type, cognitive demand, and prerequisite concepts, enabling format-specific regeneration with explicit provenance links. We report results from a tightly scoped feasibility pilot processing a single five-minute elementary STEM video offline under clean audio-visual conditions. For a predefined visual-learner profile, the system generates annotated path diagrams, two-panel instructional guides, and entity pictograms with complete back-link coverage. Ablation studies confirm individual components contribute measurably to output completeness without compromising traceability. This narrow scope precludes claims about classroom effectiveness, real-time streaming, scalability across content types, or educational impact across learner populations. We position these as testable hypotheses requiring validation across diverse content domains, authentic deployments with ambient noise and bandwidth constraints, multiple learner profiles including multilingual students and learners with disabilities, and controlled comprehension studies. The contribution is a transparent technical demonstration and methodological scaffold for investigating whether within-lesson content transformation can meaningfully support personalized learning at scale.

Keywords: multimodal learning; adaptive learning; personalized learning; virtual classrooms; speech recognition; transformer-based NLP; computer vision; Universal Design for Learning; learner modeling

1. Introduction

This study addresses the limitations of traditional one-size-fits-all virtual learning environments by proposing a Semantic Transformation and Real-Time Educational Adaptation Multimodal (STREAM) framework that leverages artificial intelligence (AI) and multimodal machine learning (ML) to enable low-latency analysis and the potential real-time delivery of personalized educational content [1]. As digital learning expands rapidly, many virtual classrooms rely on static and uniform instructional content, which fails to meet the diverse needs, preferences, and learning styles of students, especially those from underrepresented or marginalized backgrounds [2]. This study aims to bridge this critical gap by developing and validating the STREAM framework that supports analysis of instructional content and its transformation into customized formats based on individual learner profiles. Specifically, STREAM includes three key components: (1) real-time content analysis and

 doi:10.20944/preprints202510.2065.v1

knowledge extraction, (2) semantic understanding of learners' cognitive and affective characteristics, and (3) adaptive content generation and delivery through multi-modal channels such as text, audio, and video. Through a foundational survey and conceptual design, this research will lay the foundations for building intelligent virtual classrooms that dynamically respond to student variability, promoting equity, engagement, and deeper learning outcomes. STREAM also seeks to support future implementation of pilot studies and prototypes that demonstrate the feasibility and scalability of AI-driven personalized learning systems.

The COVID-19 pandemic catalyzed a global shift toward remote and hybrid learning, exposing the potential and the limitations of existing virtual education systems [3]. While many institutions rapidly adopted online platforms, the majority of these environments were designed with a uniform delivery model, failing to accommodate the diverse needs, backgrounds, and preferences of learners [4]. As a result, issues related to learner disengagement, digital fatigue, and inequitable access to meaningful learning experiences became increasingly evident [5]. In response to these challenges, educational researchers and technologists have turned to AI and emerging technologies to re-imagine instructional delivery [6]. AI provides powerful tools to transform how content is analyzed, interpreted, and adapted in real time to support personalized learning [7]. When combined with multimodal delivery methods, such as text, video, audio, and interactive media, AI can dynamically tailor instruction to learners' cognitive profiles, engagement levels, and preferred modalities [8]. Despite these advancements, real-time personalization remains largely underdeveloped in virtual learning systems. Existing models often use delayed data processing or static personalization methods that cannot respond to learners' immediate needs. This study bridges the gap between these two areas, aiming to contribute a scalable, AI-powered framework capable of real-time adaptation through multimodal ML. Doing so aligns with post-pandemic calls for more inclusive, responsive, and human-centered virtual learning environments.

Although advancements in AI and educational technologies have led to the development of various personalized learning tools, existing systems remain largely fragmented and limited in scope [9]. Current approaches to personalization often focus on isolated aspects of the learning process, such as recommending content based on prior performance or adapting quiz difficulty based on learner responses [10]. These systems rarely incorporate real-time analysis of instructional content, nor do they deliver instruction across multiple modalities—such as text, audio, and video—in a cohesive and synchronized manner [8]. Moreover, while some adaptive learning platforms integrate AI to enhance interactivity, they typically rely on pre-processed data and static learner models that cannot accommodate the dynamic and evolving nature of real-time classroom engagement [11]. Few studies explore integrating semantic communication, natural language processing (NLP), and multimodal machine learning within a unified framework for adaptive instruction. Even fewer address how such systems can function in real time to deconstruct instructional content, interpret learner preferences, and generate personalized outputs across multiple delivery formats.

This study addresses this critical gap by proposing and evaluating a holistic STREAM framework that unifies these technologies to enable real-time, multi-modal content adaptation. The goal is to go beyond piecemeal solutions and provide a scalable, end-to-end system that supports more equitable, flexible, and engaging learning experiences in virtual classrooms. This work makes three contributions. First, it establishes technical feasibility for content decomposition and format regeneration within latency bounds compatible with potential classroom use, though only under controlled conditions. Second, it provides a reproducible evaluation framework with specified metrics for recognition accuracy, tagging fidelity, processing efficiency, and output quality. Third, it offers a modular architecture compatible with Universal Design for Learning principles that can serve as a foundation for future validation studies.

This paper is organized into six main sections. Following the introduction, which presents the purpose, context, and significance of the study, each subsequent section builds a foundation for understanding and implementing the unified STREAM framework for real-time, multi-modal content adaptation in virtual classrooms:

 doi:10.20944/preprints202510.2065.v1

- Section 2: Survey of Enabling Technologies – A comprehensive review of existing AI tools and technologies relevant to real-time content analysis, learner modeling, and multi-modal delivery. The section highlights current capabilities and identifies the limitations of existing systems, underscoring the need for the proposed STREAM framework.
- Section 3: STREAM Framework: Concepts, Architecture, Components – This section introduces the conceptual framework that outlines the end-to-end flow of content from instructional source to personalized delivery. It details the system's key components, including content decomposition, learner profiling, and adaptive multi-modal presentation.
- Section 4: Feasibility and Early Prototype Design – This section presents the initial implementation strategy, including designing a pilot study using pre-recorded lectures. It outlines the methodological approach to analyzing content and simulating adaptive delivery based on the learners' preferences.
- Section 5: Discussion – An analytical discussion of how STREAM addresses existing gaps in the literature. The section examines the theoretical and practical implications of implementing such a system, with a focus on enhancing equity, engagement, and real-time responsiveness in virtual learning.
- Section 6: Conclusion and Next Steps – A summary of key contributions, followed by a roadmap for future research, including the development of subsequent papers that will explore specific components of the framework in depth.

2. Survey of Enabling Technologies

The successful implementation of a real-time multimodal adaptive learning system is based on the integration of several enabling technologies in NLP, speech recognition, CV, learning analytics, and AI-based personalization. This section surveys the current state of these technologies and identifies how they support each layer of the proposed STREAM framework.

2.1. Real-Time Content Analysis Tools

Effective real-time adaptation begins with the system's ability to accurately and efficiently analyze and deconstruct instructional content [12]. Recent advances in NLP have enabled the accurate and efficient analysis and deconstruction of instructional content, enabling the extraction of semantic information from large volumes of unstructured educational data [13]. Transformer-based models, such as BERT, T5, and GPT, are widely used to understand context, summarize content, and extract key knowledge points from text-based input, including transcripts, lecture notes, and reading materials [14]. In parallel, speech-to-text systems such as OpenAI's Whisper, Google Speech-to-Text API, and Amazon Transcribe provide robust solutions for real-time lecture transcription [15]. These tools enable the conversion of live or recorded audio into editable text, facilitating downstream semantic analysis and knowledge tagging. Integrating prosody analysis (intonation, pauses, stress) can further aid in identifying instructional emphasis and engagement cues. Video segmentation and object recognition technologies complete the multimodal picture by enabling the analysis and linking of visual elements, such as slides, gestures, and whiteboard annotations, to verbal content. Tools such as OpenCV, YOLO (You Only Look Once), and the Google Cloud Vision API support real-time detection and tracking of instructional visual content [16]. These capabilities are crucial for aligning multimodal input streams and extracting context-aware educational units.

2.2. Learner Modeling and Preference Detection

To deliver adaptive content, the system must maintain an evolving representation of each learner. Cognitive and affective modeling techniques estimate learners' attention, motivation, and emotional states by analyzing interaction data and biometric sensor data [17]. Affective computing tools, such as Microsoft's Azure Emotion API or open-source facial emotion recognition libraries, can infer frustration, confusion, or levels of interest [18]. Complementing affective data, eye-tracking and behavioral logging systems such as Tobii and iMotions monitor attention shifts, reading patterns, and

 doi:10.20944/preprints202510.2065.v1

engagement [19]. In contrast, keystroke analysis and clickstream data provide behavioral insights during learning tasks [20]. These data streams support the construction of dynamic learner profiles that evolve in real time. The system must maintain an evolving representation of each learner to deliver adaptive content. To personalize instruction, the system often relies on learning style models, such as VARK (Visual, Auditory, Reading/Writing, Kinesthetic) and Felder-Silverman's learning style dimensions (e.g., sensing/intuitive, active/reflective). Although these models are debated in the literature, they provide a practical foundation for organizing adaptive strategies and customizing content modality based on observed preferences.

2.3. Multimodal Delivery Tools

The final stage of the adaptive pipeline requires tools for generating content in different modalities. Text summarization tools such as BART and PEGASUS (Google Research) help to condense lecture transcripts into student-friendly summaries [21]. Text-to-speech (TTS) engines, such as Amazon Polly, Google WaveNet, and Microsoft Azure TTS, enable the delivery of spoken versions of content with high naturalness and emotion synthesis [22]. AI content generators such as ChatGPT, Gemini (a collaboration between Google and DeepMind), and GenAI (developed by Adobe) can create customized learning materials on demand—including explanations, quizzes, and examples—based on the student's knowledge level and learning history [23]. These tools improve the system's responsiveness and ability to provide tailored scaffolds and supports. Advanced systems may incorporate semantic communication models that optimize message transmission by focusing on meaning rather than raw data. Although still emerging, these models—utilized in domains such as edge computing and remote communication—offer promising pathways for compressing and adapting instructional content in bandwidth-constrained or distributed learning environments. (see Table 1)

Table 1. Comparison of Enabling Technologies for Adaptive Learning.

Category	Technology/Platform	Function	Limitation
Real-Time Content Analysis Tools	T5, BERT, GPT	Text-based content analysis and semantic extraction	Requires fine-tuning for educational contexts
	Whisper, Google Speech-to-Text	Speech recognition, and lecture transcription	Accuracy may drop with noisy inputs or accents
	OpenCV, YOLO, Vision API	Visual content segmentation and object recognition	Limited interpretation of abstract visuals
Learner Modeling & Preference Detection	Emotion APIs, Affective Computing Tools	Detects emotional and motivational states in learners	Potential bias; limited granularity without hardware
	Eye-tracking (Tobii, iMotions)	Tracks gaze, attention, and behavioral engagement	Intrusive or costly; sensitive to setup
	VARK, Felder-Silverman Models	Categorizes learners by preferred learning modalities	Contested theoretical validity
Multimodal Delivery Tools	TTS Engines (Polly, WaveNet)	Delivers content in natural spoken formats	Modality fidelity varies by language and platform
	ChatGPT, Gemini, GenAI	Generates custom content for adaptive instruction	Limited control over depth and granularity
	Semantic Communication Models	Optimizes message meaning in low-bandwidth settings	Still emerging; high technical complexity
Existing Adaptive Learning Systems	Khan Academy, Coursera, Smart Sparrow	Personalized paths based on performance history	Lacks real-time adaptation and multimodal personalization

2.4. Existing Adaptive Learning Systems

Several platforms have pioneered adaptive learning, including Khan Academy, Coursera, Duolingo, ALEKS, and Smart Sparrow. These systems provide personalized learning paths tailored to user input and assessment data. For example, Khan Academy recommends exercises based on previous performance. It has integrated AI tools such as Khanmigo for real-time feedback, while Coursera uses reinforcement learning to sequence courses and modules [24,25]. Duolingo employs adaptive gamified algorithms for language learning, adjusting difficulty based on user responses, and ALEKS applies knowledge-space theory to personalize math [26]. However, these platforms are primarily based on predefined adaptation rules and static learner models, lacking the capacity for

 doi:10.20944/preprints202510.2065.v1

real-time multimodal personalization [27]. They often do not incorporate affective data, seamless multimodal delivery, or real-time feedback loops that adapt during live sessions. Moreover, their adaptive strategies are typically limited to text, video, or quiz formats, with minimal support for switching between modalities or dynamically adjusting content based on emotional or behavioral cues in virtual classroom settings [28–30]. This gap highlights the need for a more comprehensive, AI-driven framework that integrates semantic understanding, real-time data analysis, and multimodal content generation to support a truly personalized and inclusive learning experience (see Figure 1).

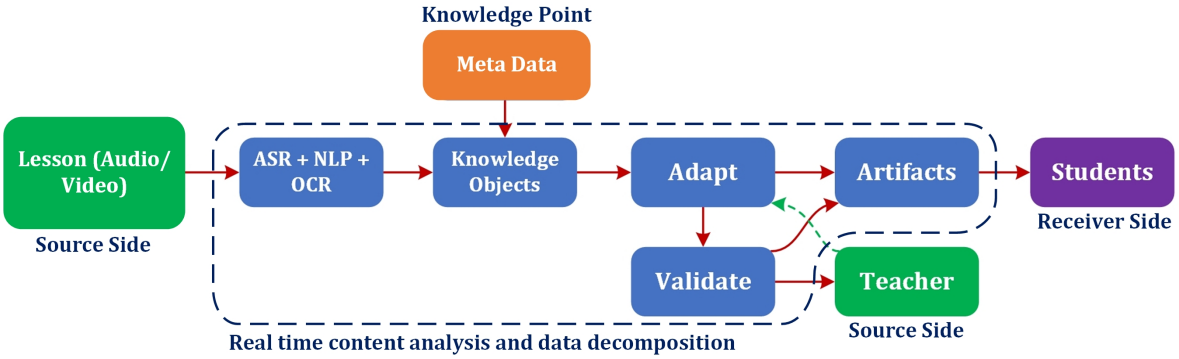

Figure 1. Study Diagram for Adaptive Learning System

To further illustrate these limitations and position the proposed STREAM framework, Table 2 provides a detailed comparison with additional contemporary systems, drawing from recent 2025 reviews [31,32]. This table highlights STREAM's unique emphasis on real-time, AI-powered transformation within the instructional flow, which existing systems essentially treat as static or post-hoc. For instance, while Khan Academy excels at accessible, performance-driven paths, it does not decompose a live lecture's speech and visuals on the fly to create personalized artifacts like tagged diagrams. Similarly, Coursera's sequencing is robust but lacks multimodal regeneration to enable immediate alignment among learners.

Table 2. Comparison of the STREAM Framework to Existing Adaptive Learning Systems.

Feature	STREAM	Khan Academy	Coursera	Smart Sparrow	Duolingo	ALEKS
Real-Time Adaptation	Yes: Decomposes and adapts content during live/streamed lessons with <1 s latency on standard hardware.	Partial: AI feedback via Khanmigo; adapts between exercises from post-performance data; limited in-lesson processing.	Partial: Recommends modules post-quiz; no live decomposition/regeneration.	Partial: Adaptive simulations, but rule-based and not real-time for all content types.	Partial: Adjusts difficulty in-session, but limited to gamified drills without full content transformation.	No: Adapts paths based on assessments; offline processing dominates.
Multimodal Content Delivery	Yes: Dynamically generates/regenerates across text, audio, video, diagrams; fuses ASR/NLP/CV for seamless integration.	Partial: Text, video, exercises with some AI narration; no real-time modality switching or generation.	Partial: Video lectures, quizzes, text; limited to pre-made formats without fusion.	Yes: Interactive simulations with text/video; not AI-driven regeneration for live contexts.	Partial: Audio/text drills, images; app-based, no video decomposition or custom generation.	No: Primarily text-based math problems; minimal multimodal support.

 doi:10.20944/preprints202510.2065.v1

Table 2. *Cont.*

Feature	STREAM	Khan Academy	Coursera	Smart Sparrow	Duolingo	ALEKS
Personalization Depth	High: Dynamic learner profiles (cognitive/affective states via eye-tracking, emotion APIs); adapts to preferences, disabilities, multilingual needs with UDL compatibility.	Medium: Performance-based paths with AI tutoring; basic mastery tracking, limited affective or real-time behavioral modeling.	Medium: Skill-based recommendations; learner profiles limited to progress/history.	High: Scenario-based adaptation; includes some behavioral cues, but not deeply affective.	Medium: Skill/decay models; gamified, but no deep affective or disability-focused profiles.	High: Knowledge-space theory for math; detailed but domain-specific; no multimodal/affective.
Content Decomposition & Tagging	Yes: AI-driven (BERT/T5 for semantics, Whisper for speech, YOLO/OpenCV for visuals); tags units with metadata for traceability.	No: Relies on pre-tagged content; no automated decomposition.	No: Courses are pre-structured; no real-time tagging.	Partial: Tags simulations; manual/author-driven, not AI-automated.	No: Pre-built lessons; algorithmic but not decomposed via multimodal AI.	No: Pre-defined knowledge points; no real-time multimodal tagging.
Equity & Accessibility Focus	High: Designed for diverse populations; supports multilingual use and disabilities via regenerated formats and provenance links.	Medium: Free access, subtitles, AI for underserved areas; largely one-size-fits-all.	Medium: Subtitles, mobile access; partnerships for equity, but not adaptive regeneration.	Medium: Customizable for inclusivity; deployment-limited.	High: Multilingual support, gamification for engagement; app-centric, less for disabilities.	Medium: Adaptive pacing; limited multimodal accessibility.
Scalability & Hardware Needs	High: Modular pipeline for classroom-grade hardware; pilot-tested in clean conditions with a roadmap for noisy/bandwidth-constrained extensions.	High: Web/app-based; scales globally.	High: Cloud-based; accessible worldwide.	Medium: Requires authoring tools; less scalable for non-experts.	High: Mobile-first; scales via app ecosystem.	High: Web-based; LMS integrations; math-focused.
Validation & Evidence	Pilot-based: Feasibility on a 5-minute STEM clip; staged roadmap for diverse testing (e.g., multilingual, disabilities).	Extensive: Data from millions; A/B tests on mastery learning and AI efficacy.	Extensive: University partnerships; completion-rate analyses.	Research-backed: Studies on adaptive simulations.	Extensive: App metrics; language-retention studies.	Research-backed: Knowledge-space model validated in education studies.

3. STREAM Framework: Concepts, Architecture, Components

3.1. Conceptual Flow

3.1.1. Source Side

The source side of STREAM represents the origin of instructional input within a virtual or hybrid learning ecosystem. It encompasses live teaching sessions, pre-recorded lectures, multimedia

 doi:10.20944/preprints202510.2065.v1

instructional materials, and other educational content delivered by instructors or digital platforms. In current virtual learning models, such content is often static and generalized, offering the same instructional experience to all students regardless of their individual needs or learning preferences. This study begins by recognizing the diversity of instructional materials as a rich source of adaptable content that, if properly analyzed and deconstructed, can serve as the foundation for personalized learning experiences. At the heart of this stage is the human or AI-assisted teacher, who generates the educational narrative through verbal explanations, slide presentations, and embedded media such as images or video clips. When captured in real-time or provided as recordings, these materials hold key semantic and structural components necessary for downstream content analysis. In a traditional setting, the teacher controls the pace, emphasis, and interaction; however, in virtual settings, these features are not always effectively captured or utilized to support different learners [33]. STREAM acknowledges the value of teacher-delivered content not only as instructional input but also as a data-rich source for real-time feature extraction.

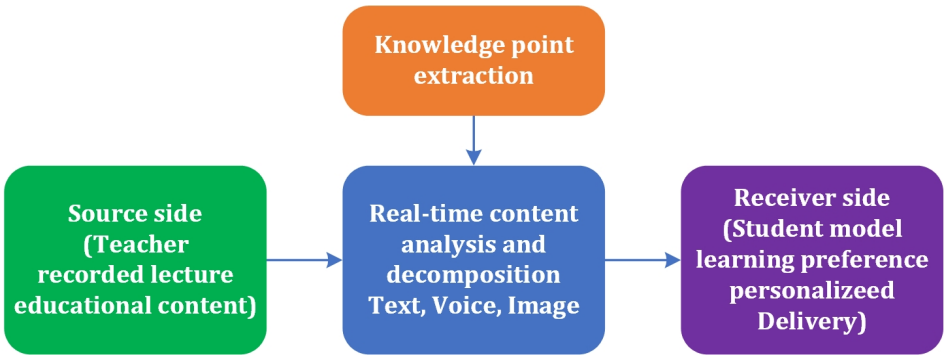

Figure 2. Conceptual Flow of the Proposed STREAM Framework.

In addition, recorded lectures provide opportunities for asynchronous engagement, which can be particularly beneficial for students who require flexible access or repeated exposure to instructional material. However, without adaptive elements, these recordings remain passive and non-responsive [34]. This framework re-imagines recorded lectures as analyzable entities whose content can be broken down using AI-powered tools such as speech-to-text engines, video segmentation algorithms, and semantic parsers. These tools enable the identification of key learning units, transitions, tone changes, and pedagogical markers that may otherwise be lost in static video playback [35]. The final component on the source side is the broader category of educational content, which includes digital textbooks, interactive simulations, annotated slides, quizzes, and multimedia supplements [36]. These materials often exist in isolated formats, lacking interoperability or cohesive integration with adaptive learning systems [37]. The STREAM framework addresses this challenge by treating all educational content as modular inputs that can be semantically indexed and transformed to meet the learner's needs. This modular view enables the system to extract knowledge points, generate metadata, and feed content into the next stage—real-time analysis and learner-centered adaptation—laying the groundwork for truly personalized and multi-modal virtual classrooms.

3.1.2. Middle Layer

The middle layer of STREAM serves as the intelligent processing core, transforming static instructional materials into dynamically adaptable components. This stage leverages advanced AI techniques, including NLP, speech recognition, and computer vision (CV), to perform real-time analysis and decomposition of educational content across multiple modalities [38]. This layer aims to extract meaningful pedagogical elements from text, voice, and image data, enabling the system to understand the structure, semantics, and instructional intent behind each piece of content. In the text domain, real-time content analysis involves parsing lecture transcripts, digital documents, and on-screen annotations to identify key knowledge points, learning objectives, and pedagogical cues [39]. Using transformer-based language models such as Bidirectional Encoder Representations from Trans-

 doi:10.20944/preprints202510.2065.v1

formers (BERT), Text-to-Text Transfer Transformer (T5), or GPT, the system can recognize conceptual hierarchies, thematic transitions, and emphasis markers (e.g., definitions, examples, or summaries) [40]. This process enables the classification of content by difficulty, relevance and instructional purpose, supporting the generation of personalized instructions and scaffolded learning sequences tailored to different types of learners [41].

In the voice mode, speech-to-text technologies such as OpenAI's Whisper or Google's Speech Application Programming Interface (API) are employed to transcribe live or recorded lectures in real time [42]. Beyond transcription, STREAM also analyzes speech prosody—including tone, pauses, and emphasis—to identify points of instructional stress or learner confusion. For example, a teacher's repeated phrasing or slower articulation may indicate important concepts that personalized outputs should highlight or reinforce. These insights improve the system's ability to prioritize content and respond adaptively based on perceived educational importance [43]. Meanwhile, image and video decomposition are critical in capturing non-verbal instructional cues [44]. Visual content, such as lecture slides, diagrams, screen annotations, and gestures, can be analyzed using CV techniques, including object detection, optical character recognition (OCR), and image segmentation [45]. These tools allow the system to isolate figures, extract textual elements from visuals, and map visual themes to accompanying spoken or written explanations [46]. This multi-modal fusion enriches the content representation and ensures that learners with varying modality preferences (e.g., visual vs. auditory learners) receive content in forms that align with their strengths. Overall, this middle layer processes and decodes complex instructional input, reorganizing it into a structured, machine-interpretable format. Doing so creates a semantic bridge between teacher-driven content and the learner-facing adaptive delivery mechanisms. This stage is crucial for enabling real-time responsiveness and personalization, ensuring that every element passed on to the learner is relevant, meaningful, and optimally aligned with their unique learning profile.

3.1.3. Receiver Side

The receiver side of the STREAM framework is focused on the learner, the ultimate beneficiary of adaptive content transformation. This layer involves developing a dynamic student model that captures individual learning preferences, cognitive styles, affective states, and engagement patterns [47]. STREAM aims to foster a highly personalized educational experience that enhances comprehension, retention, and motivation among diverse student populations by aligning instructional delivery with these learning-specific attributes. At the core of this process is the student model, a real-time profile that evolves based on continuous interaction data, such as content engagement, response accuracy, and modality preference. This model includes static attributes (e.g., preferred learning style, language proficiency, accessibility needs) and dynamic attributes (e.g., emotional state, attention span, pace of progress). The system draws on input from behavioral logging, biometric sensors (if available), embedded assessments, and user interactions to accurately assess these dimensions. Integrating affective computing and machine learning algorithms makes the student model increasingly refined, enabling a more nuanced understanding of learner variability.

Once a complete profile of the learner is established, the system proceeds to personalized delivery, where the instructional content is tailored to suit the format, pace, and complexity of the learner [48]. For instance, a visual learner might receive an infographic instead of a textual explanation, while an auditory learner might receive the same content as a narrated summary [49]. Similarly, students who struggle with certain concepts may be offered simplified definitions, scaffolded tasks, or additional examples to support their understanding [50]. The metadata and knowledge components extracted in the middle layer enable this content adaptation, allowing the system to match the instructional intent with the needs of the learner. Importantly, the system's multi-modal capabilities ensure that content can be delivered across various channels—text, speech, video, or even interactive simulations—depending on the learner's preferences and context. This flexibility is critical for inclusive design, particularly for students with disabilities, language barriers, or non-traditional learning trajectories. In addition, personalized feedback and progress tracking are built into the delivery system, allowing learners to

 doi:10.20944/preprints202510.2065.v1

monitor their development and engage in self-regulated learning behaviors. In sum, the receiver side operationalizes the pedagogical goal of personalization by dynamically mapping processed content to individualized learner profiles. It represents a significant departure from conventional e-learning systems that treat all learners uniformly, offering a scalable, AI-powered approach to responsive, multi-modal, and learner-centered education instead.

3.2. Key Components

3.2.1. Knowledge Point Extraction

The extraction of knowledge points is a foundational component of the STREAM framework, serving as the mechanism to identify and isolate the core instructional concepts from various educational inputs [51]. These "knowledge points" refer to discrete units of learning—such as definitions, formulas, ideas, skills, or relationships—that represent the fundamental building blocks of academic content [52]. In traditional instruction, teachers emphasize these points through verbal cues, textual highlights, or visual annotations [53]. However, in a digital learning context, especially one mediated by AI, identifying and extracting these points must be automated and contextually aware [54]. This extraction process begins with the semantic analysis of multimodal input, including text from lecture transcripts, spoken explanations, annotated slides, and visual materials. NLP techniques are used to detect linguistic structures commonly associated with instructional emphasis, such as transitional phrases (e.g., “the key takeaway is…”), definitions (“X is defined as…”), or cause-and-effect structures. Machine learning models trained on educational datasets can be employed to classify sentences or segments into instructional categories (e.g., explanation, elaboration, example, conclusion), making it easier to pinpoint high-priority knowledge units.

In addition to semantic cues, structural features—such as font size, bold text, bullet points, and slide titles—are leveraged through CV and OCR tools when processing visual content [55]. These features provide additional layers of meaning that help distinguish primary concepts from supporting details. The system may also use domain-specific ontologies or concept maps to cluster related knowledge points and establish hierarchical relationships, thereby enhancing coherence and instructional flow during adaptive delivery. Once extracted, these knowledge points are stored as modular data objects, each tagged with metadata indicating its source, difficulty level, instructional purpose, and preferred modality. This structure enables the system to reassemble and repack content flexibly to suit different learner profiles. For example, a single concept may be transformed into a concise text summary for a proficient reader, a narrated animation for a visual auditor, or a series of scaffolded instructions for a student requiring additional support [56]. Ultimately, knowledge point extraction transforms passive instructional content into active, machine-readable units that can drive intelligent personalization. Enhances the reusability and modularity of the content, ensuring that personalization remains pedagogically meaningful rather than superficial. This process engages diverse learners through multimodal engagement.

3.2.2. Metadata Generation

Metadata generation is critical to bridge raw instructional content with intelligent personalization by enriching extracted knowledge points with descriptive and contextual information [1]. In this context, metadata refers to structured data that describes the attributes, purpose, and pedagogical context of a given content unit—such as its learning objective, difficulty level, compatibility of modes, emotional tone, and cognitive demand [57]. This abstraction layer provides meaningful adaptation tailored to individual learner needs. The process begins immediately after the extraction of knowledge points, where each segment of content is tagged with pedagogical metadata. This includes categorical information such as the content type (e.g., definition, explanation, example), cognitive level based on Bloom’s Taxonomy (e.g., remember, apply, analyze), and alignment with curriculum standards or learning outcomes [58]. AI models trained on annotated educational corpora can automatically infer these labels by analyzing syntax, semantics, and contextual cues [59]. Additionally, heuristics or

 doi:10.20944/preprints202510.2065.v1

predefined instructional design rubrics generate instructional metadata, such as estimated learning time, required prerequisites, and ideal delivery modality (e.g., text, video, animation) [60].

In addition, affective and engagement-related metadata can also be attached to content, particularly when analyzing speech or video input [61]. For example, speech prosody analysis might detect moments of excitement or emphasis in a teacher's voice, indicating emotionally charged or critical instructional moments [62]. Depending on a student's emotional profile or engagement patterns, these markers can be flagged for later emphasis or moderation [63]. Similarly, visual cues such as teacher gestures, slide animations, or on-screen annotations can be encoded as attention indicators, guiding the system on where to focus adaptive feedback [64]. All generated metadata is stored in a content repository as part of a modular object structure, allowing the system to be customized to the needs of the real-time learner [65]. For example, suppose a student struggles with a particular concept. In that case, the system can use metadata tags to locate a simpler explanation, visual reinforcement, or a real-world example from the content library. In contrast, the same metadata can be used to provide advanced learners with enriched higher-order content. In essence, metadata generation transforms educational content from static assets into flexible, searchable, and pedagogically-aware units, making real-time personalization scalable and pedagogically sound [66]. It enables the adaptive system to reason about both content and the learner in a structured manner, ensuring that the delivery is not only customized but also contextually and cognitively appropriate.

3.2.3. Learner Profiling

Learner profiling is the adaptive engine's intelligence layer that personalizes instruction based on each student's unique characteristics, preferences, and learning behaviors [67]. This component constructs a dynamic, data-driven profile that captures the cognitive, behavioral, emotional, and contextual attributes of the learner [68]. The profile is continuously updated as the student interacts with the system, allowing real-time content adjustments that promote engagement, comprehension, and long-term retention [1]. At its foundation, the learner profile begins with static attributes, such as age, language proficiency, previous academic performance, and declared learning preferences (e.g., visual, auditory, kinesthetic) [69]. These are often collected through initial diagnostic surveys or onboarding modules. While helpful in forming a baseline, static profiles alone are insufficient for adaptive learning at scale. Therefore, the system also generates and updates dynamic attributes, which evolve based on the real-time interactions of the learner. These include time-on-task, response latency, precision, preferred mode of engagement during engagement, emotional state (if detected by affective computing), and feedback response patterns [70]. To model these attributes meaningfully, the system utilizes machine learning algorithms that interpret clickstream data, biometric feedback (e.g., eye movements, facial expressions, if available), quiz performance, and content interaction logs. These inputs feed into a continuously evolving learner model that predicts not only what the learner knows, but also how they learn best and how their needs may change over time or in different contexts. For instance, if a student frequently pauses during video lectures but excels with interactive graphics, the model adapts to prioritize visual learning aids in future sessions.

In advanced implementations, affective and motivational states are also captured using real-time sensors or AI-based emotion recognition tools [71]. For example, if a student shows signs of frustration or disengagement, the system may intervene with encouraging messages, alternate modalities, or content simplification to maintain motivation [72]. Over time, these interventions are tracked and used to refine the learner profile further, forming a feedback loop between the learner's emotional journey and instructional adaptation. The ultimate function of a learner profile is to inform content selection, sequence, and delivery style, matching instructional components—tagged with rich metadata—to learners in a way that optimizes cognitive alignment and emotional resonance. It ensures that each learner receives a pathway through content that is not only academically appropriate but also motivationally and contextually supportive. In sum, learner profiling transforms the adaptive system from a reactive tool to a proactive and anticipatory tutor, capable of delivering real-time, learner-centered instruction that evolves with each student's growth [73]. This component is central to

 doi:10.20944/preprints202510.2065.v1

achieving STREAM's vision of inclusive, personalized virtual classrooms that recognize and respond to diverse student needs.

3.2.4. Adaptive Content Generation

Adaptive content delivery is the culminating component of STREAM, where analyzed instructional material, enriched with metadata and guided by learner profiles, is transformed into a personalized learning experience tailored to each student's preferences, needs, and cognitive profile [74]. This stage operationalizes the promise of real-time, AI-driven instruction by dynamically modifying what content is delivered, how it is presented, and when it is introduced, thus aligning educational experiences with learners' pathways. At its core, adaptive delivery uses the knowledge points and metadata extracted in previous layers and matches them to the evolving learner profile of the student [75]. Based on this match, the system determines the most effective modality and format for each learning unit. For example, a student who demonstrates strong verbal comprehension but lower visual processing speed might receive a narrated explanation paired with simple, high-contrast visuals rather than dense diagrams [76]. In contrast, students with high visual-spatial intelligence might be offered interactive infographics or simulations instead of textual content [77]. This personalization is powered by real-time decision algorithms that select from a repository of modular instructional assets, each tagged with pedagogical, cognitive, and emotional metadata [78]. These assets are assembled into adaptive learning sequences, where the content is adjusted in form and complexity, as well as pacing, sequencing, and scaffolding. If a student struggles with a particular topic, the system might interject with simpler explanations, offer immediate practice opportunities, or revisit prerequisite knowledge before progressing. If students excel, the system can skip redundant material and introduce more challenging enrichment-oriented tasks.

Importantly, adaptive delivery is multimodal by design, supporting the integration of text, video, audio, animations, haptic interactions, and gamified elements, whichever modes best align with the context of the learner [8]. This multimodality is crucial for inclusivity, as it supports students with sensory impairments, learning disabilities, or diverse cultural backgrounds. It also allows the system to adapt flexibly to the environment, such as switching from visual to auditory delivery in mobile or low-bandwidth settings. Another essential feature of this component is continuous feedback and real-time monitoring. As students engage with personalized content, the system dynamically tracks engagement metrics and performance indicators to adjust future content. This creates a looped system where delivery is informed by performance and performance informs delivery, making learning experiences not only personalized but responsive. Ultimately, adaptive content delivery actualizes STREAM's vision of a virtual classroom centered on the learner, where each student receives just-in-time instruction tailored to their learning style, pace, and emotional readiness. It closes the loop of the STREAM framework by transforming abstract data into concrete, impactful educational experiences, positioning learners not as passive recipients of information but as active participants in a personalized and adaptive learning journey. (see Table 3)

 doi:10.20944/preprints202510.2065.v1

Table 3. Summary of Key Components in the Adaptive Learning Framework.

Component	Purpose	Technologies Used	Role in Framework	Conceptual Flow Location
Knowledge Point Extraction	Identifies and isolates core instructional concepts (definitions, skills, etc.) from multimodal content.	Transformer-based NLP (e.g., BERT), OCR, semantic parsing.	Converts instructional content into modular, meaningful learning units.	*Middle layer* (content analysis / decomposition)
Metadata Generation	Adds descriptive and pedagogical tags (e.g., type, difficulty, modality) to enable intelligent retrieval and alignment.	Heuristic tagging, Bloom's taxonomy mapping, prosodic/emotional analysis.	Provides a metadata layer for content organization and adaptive use.	*Middle layer* (content analysis / decomposition)
Learner Profiling	Builds dynamic profiles based on learner preferences, behaviors, and emotional states to guide personalization.	Machine learning, affective computing, behavioral analytics.	Guides decision-making on what and how to present content.	*Receiver side* (student model)
Adaptive Content Delivery	Delivers content in customized formats and sequences across modalities, adapting in real-time to learner responses.	Decision algorithms, multimodal rendering engines, real-time feedback loops.	Implements learner-facing adaptations for engagement and mastery.	*Receiver side* (personalized delivery)

3.3. Modularity

A key strength of the proposed adaptive learning STREAM framework lies in its modular design, which allows each component—source input, content analysis, learner modeling, and adaptive delivery—to function as a standalone system while contributing to the holistic objective of personalized instruction in real-time [75]. The source side of STREAM, including teacher-led instruction, recorded lectures, and digital content, offers opportunities to study how various modalities of instructional delivery (e.g., live vs. recorded, structured vs. informal) affect the fidelity and richness of the content available for real-time analysis [79]. The middle layer of real-time content analysis and decomposition contains subtopics that warrant individual investigations [80]. One possible study could focus exclusively on text-based decomposition using transformer models, while another might explore voice-based prosody analysis to detect instructional emphasis. A third option could investigate CV techniques to extract semantic elements from visual content, such as lecture slides or whiteboards.

The student modeling component opens the door to in-depth research on learner analytics, including affective computing, cognitive state prediction, and real-time behavioral profiling [17]. Each of these subdomains could serve as a standalone empirical study, especially when combined with methods such as eye-tracking, engagement logging, or sentiment analysis during instructional sessions. Ultimately, adaptive content delivery encompasses a broad and impactful area of research, including personalized modality selection, learning path generation, and multimodal feedback systems [81]. Researchers could conduct experiments comparing learners' results across different adaptive strategies or studying the effectiveness of modality switching based on emotional or cognitive indicators. By designing the STREAM framework as a set of discrete, interlinked modules, this research agenda enables iterative development, targeted validation, and cross-disciplinary collaboration. Each module is a functional building block of the overall system and a fertile ground for scholarly inquiry capable of generating its literature, tools, and pedagogical implications. This modular structure positions STREAM as a scalable blueprint for applied development and theoretical advancement in AI-driven, learner-centered education.

4. Feasibility and Early Prototype Design

To evaluate the viability of the STREAM framework, we implement a deliberately scoped, end-to-end prototype that runs the full content–to–adaptation loop on a short, pre-recorded elementary STEM lesson. The pipeline operates *inside* the instructional stream: teacher-delivered explanations are transcribed, semantically tagged, and (where available) temporally aligned to on-screen visuals to produce machine-interpretable knowledge objects with provenance and pedagogical metadata. These objects are then regenerated as learner-aligned artifacts—in this pilot, visual-first diagrams and panels for a predefined "visual" profile—thereby demonstrating real-time content transformation rather than post hoc recommendation or remixing.

 doi:10.20944/preprints202510.2065.v1

The prototype is intentionally minimal, as it processes a single five-minute "Pree" clip using directional logic and applies rule-based adaptation for one learner profile, prioritizing feasibility over feature completeness. Success is defined by whether the full pipeline can execute on commodity hardware at classroom-scale latency while preserving traceability from adapted outputs back to their source timecodes. This framing yields qualitative feasibility evidence and timing logs (rather than summative learning outcomes), positioning subsequent sections to specify and report accuracy, latency, and readability/traceability criteria.

This pilot (1) *extract* instructional elements—definitions, step sequences, prompts, and contextual entities—from a short, multimodal lesson by combining time-aligned ASR and transformer tagging under a compact label schema to support reliable validation, (2) *transform* those elements into a visual-first representation for a single predefined learner profile (visual/VARK) using rule-based mappings (e.g., arrow-sequence → step-numbered diagram; prompt → two-panel "Plan→Test" guide; entity → pictogram with OCR-derived captions), and (3) accomplish both with practical latency on commodity hardware (single-Graphics Processing Unit (GPU) workstation) so the full pipeline operates at classroom-scale speeds. Success is operationalized by measurable criteria specified in the following section, including ASR quality, tagging fidelity, end-to-end time, and output readability/traceability. These Components are aligned to Table 3. This feasibility pilot exercises the four components as follows:

(i) Knowledge Point Extraction — time-aligned ASR and transformer tagging produce machine-interpretable objects for definitions, step sequences, prompts, and entities;

(ii) Metadata Generation — each object carries provenance and pedagogical fields (ID, timecodes, Bloom level, difficulty, prerequisites, visual references) to preserve auditability;

(iii) Adaptive Content Delivery — rule-based mappings regenerate visual-first artifacts (arrow-sequence → numbered path diagram; prompt → two-panel *Plan→Test*; entity → pictogram with OCR caption);

(iv) Learner Profiling — intentionally out of scope in this pilot (fixed visual profile) to isolate feasibility of the content–to–adaptation loop.

Metric-to-component keys. ASR WER and tagging κ (Extraction); provenance/readability checks (Metadata); end-to-end latency/resources and artifact quality (Delivery).

4.1. Source Side

4.1.1. Pre-Recorded Lecture (e.g., from Pree")

The pilot draws on a set of pre-recorded, coding-focused mini-lessons delivered by an early-elementary STEM facilitator ("Pree"). Each lesson is designed to approximate a classroom exchange: the facilitator narrates brief explanations, elicits short peer responses, and uses simple, high-contrast visuals—such as arrow cards, toy vehicles, and destination boards—to keep linguistic cues and visual symbols co-present within short, well-bounded scenes. For this study, we focus on a *five-minute* segment on elementary path planning that extends a previously taught repertoire by introducing a *right-turn* arrow alongside an existing *forward* arrow. Within this clip, learners compose and test action sequences to reach named destinations (e.g., a farm, zoo, or shopping center). The combination of clear narration and distinct on-screen symbols produces repeated, easily identifiable primitives, such as tokens (`forward` and `right`), destination labels, and arrow glyphs, yielding stable anchor points for temporal alignment and provenance tracking. (See Figure 3).

The selection of this segment was intentional. Pedagogically, the interaction follows a concise and widely used pattern—concept introduction, guided practice, and brief collaborative activity—creating natural boundaries for segmentation, time-aligned tagging, and later audit. Multimodally, the co-occurrence of speech, symbolic arrows, and labeled boards enables cross-validation across modalities: automatic speech recognition (ASR) captures verbal prompts and step language; CV and OCR recover arrow orientation and destination text; and a semantic tagger organizes the discourse into definitions, steps, prompts, and entities. Signal quality is deliberately clean: the audio track is uncomplicated, and the visual symbols are high-contrast and visually distinct, which improves ASR robustness and

 doi:10.20944/preprints202510.2065.v1

simplifies glyph/label detection for a lightweight vision pass while still supporting the construction of traceable knowledge objects. Finally, the compact five-minute duration satisfies latency and resource constraints on commodity hardware without sacrificing representativeness: the clip includes multiple destinations and sequence variations, ensuring sufficient complexity for an end-to-end feasibility test.

Figure 3. The Coding Class

Operationally, the clip provides three classes of inputs directly consumable by the pipeline. First, it provides *step sequences* (e.g., `forward, forward, right`) that can be parsed, canonicalized, and later rendered as numbered path diagrams. Second, it contains *instructional prompts* (e.g., "show me the path," "test it with your car") that the imperative detector can identify and map to two-panel *Plan→Test* guides. Third, it includes *contextual entities* (such as vehicles and destinations) that can be linked to pictograms and OCR-derived captions. Together, these properties make the source stream both analytically tractable and pedagogically meaningful, providing a controlled yet authentic substrate on which to evaluate the feasibility of the proposed content-to-adaptation loop.

Example (Illustrative transcript for three input classes). ASR-normalized excerpt showing step sequences, instructional prompts, and contextual entities.

> **S1** [00:02:10–00:02:15]: From the start, [STEP]`forward, forward, right`[/STEP] to reach the [ENTITY]zoo[/ENTITY].
> **S2** [00:02:15–00:02:17]: [PROMPT]Show me the path[/PROMPT].
> **S3** [00:02:17–00:02:19]: [PROMPT]Test it[/PROMPT] with your [ENTITY]car[/ENTITY].

Interpretation: [STEP] canonical arrow sequence used for numbered path diagrams; [PROMPT] imperative mapped to a two-panel *Plan→Test* guide; [ENTITY] destination/vehicle linked to a pictogram and (when available) OCR caption. This excerpt aligns with the pilot's schema and renderer: step sequences (e.g., `forward, forward, right`) render as path diagrams; imperatives (e.g., "show me the path," "test it…") render as *Plan→Test* guides; entities (e.g., "zoo," "car") materialize as pictograms with OCR-derived captions.

 doi:10.20944/preprints202510.2065.v1

4.2. Middle Layer

4.2.1. Content Component Extraction

In the middle layer, the lesson is decomposed into machine-readable instructional units through a three-stage pipeline—ASR → → NLP → → planned CV—that yields timestamped, semantically tagged spans traceable to specific audiovisual evidence. Each stage records confidence scores and timing metadata, allowing downstream adaptation to prioritize high-certainty items and, where necessary, defer to rapid human confirmation. This design preserves end-to-end auditability while supporting practical latency on commodity hardware.

Speech-to-text. We employ Whisper with overlapping 20-second windows (5-second stride) and energy-based voice activity detection (VAD) to stabilize timestamps during rapid turn-taking. The decoder returns token- and utterance-level times; punctuation and casing are restored, and a light normalization pass removes fillers (e.g., *um/uh/like*), standardizes numerals (e.g., "three rights" → → `3 right`), and harmonizes measurement phrases. Speaker turns are preserved at utterance boundaries to retain emphasis timing (for example, slower repetitions during concept introduction). For quality control, the ASR stage emits per-segment confidence scores and a compression-ratio flag to surface likely garbles for targeted manual spot-checks in the pilot.

Semantic tagging. We illustrate sentence segmentation, multi-label prediction, rule-assisted imperatives, and arrow-sequence canonicalization on a short ASR-normalized snippet. The final tag items export per-label probabilities and evidence spans (0-indexed, end-exclusive), enabling audit and confidence-aware filtering downstream.

Example (ASR transcript). Normalized; three sentences:

```
S1 [00:02:10.00-00:02:15.00]:
  "From the start, go forward, forward, then right to reach the zoo."
S2 [00:02:15.10-00:02:16.80]:
  "Show me the path."
S3 [00:02:17.00-00:02:18.50]:
  "Test it with your car."

Sentence segmentation (spaCy)
[ S1 | S2 | S3 ]

Multi-label classifier outputs (BERT + sigmoid)
# probs shown only for labels >= 0.05
S1:
  knowledge_point: 0.88
  entity:          0.11   (token ‘‘zoo’’)
  example:         0.07
S2:
  prompt:          0.95
S3:
  prompt:          0.93

Rule-assisted passes
Imperative detector:
  S2 → prompt=True  (verb-initial ‘‘Show’’)
  S3 → prompt=True  (verb-initial ‘‘Test’’)
Arrow-sequence parser (token collapse):
  S1 evidence span chars [16,47) = "go forward, forward, then right"
  canonical_steps: ["forward","forward","right"]
```

 doi:10.20944/preprints202510.2065.v1


```
Exported tag items (JSON Lines)
"sentence_id":"S1",
"time":{"start":"00:02:10.00","end":"00:02:15.00"},
"text":"From the start, go forward, forward, then right to reach the zoo.",
"labels":{"knowledge_point":0.88,"entity":0.11,"example":0.07},
"evidence_spans":{"steps":[16,47]},
"canonical_steps":["forward","forward","right"]

"sentence_id":"S2",
"time":{"start":"00:02:15.10","end":"00:02:16.80"},
"text":"Show me the path.",
"labels":{"prompt":0.95},
"evidence_spans":{"prompt":[66,83]}

"sentence_id":"S3",
"time":{"start":"00:02:17.00","end":"00:02:18.50"},
"text":"Test it with your car.",
"labels":{"prompt":0.93},
"evidence_spans":{"prompt":[84,106]}
```

Interpretation: With a default threshold of $p \geq 0.70$, S1 yields a `knowledge_point` with a canonicalized step string, and S2–S3 yield `prompt` items. These flow into packaging as knowledge objects with IDs/timecodes and then render for visual learners as a numbered path diagram (from `canonical_steps`) and a two-panel *Plan*→*Test* guide (from `prompt`).

Planned visual alignment. CV modules are introduced incrementally and scoped to the pilot. First, OCR (Tesseract) runs over high-contrast regions to recover destination labels (e.g., "zoo," "shopping"). Second, OpenCV contour- and shape-based heuristics detect arrow glyphs and estimate orientation via principal-axis analysis. Third, a weak cross-modal alignment links visual detections to co-occurring transcript spans within a ±2±2-second window. When multiple candidate frames compete for a span, the system selects the track with maximal intersection-over-union to maintain temporal consistency. Low-confidence or occluded symbols are surfaced to a lightweight human-verification UI (checkbox confirmation) rather than being accepted automatically, preserving both speed and traceability.

Knowledge-object packaging. All extracted elements are consolidated into traceable knowledge objects that carry pedagogical metadata and provenance. Core fields include

```
"id",
"timecodes",
"text",
"visual_refs",
"type" {definition, step, prompt, example},
"bloom_level",
"difficulty",
"prerequisites"
```

and, for feasibility analysis and reproducibility, the pilot also records `asr_conf`, `tag_conf_map`, `ocr_conf`, `bbox` ($[x, y, w, h]$ in pixels for each reference frame), and a short `source_hash` of the media segment. Timecodes are given as `hh:mm: ss.xx` strings. Below, we show representative objects.
Example (step). A canonicalized arrow sequence aligned to a short span:

```
"id": "KO-0142",
"timecodes": {"start": "00:02:11.20", "end": "00:02:14.90"},
"text": "forward, forward, right",
```

 doi:10.20944/preprints202510.2065.v1


```
"type": "step",
"asr_conf": 0.93,
"tag_conf_map": {"knowledge_point": 0.88},
"visual_refs": [{"frame": 3187, "bbox": [412, 276, 86, 44], "ocr": null}],
"bloom_level": "apply",
"difficulty": "intro",
"prerequisites": ["KO-0061: forward arrow meaning"],
"source_hash": "c7a9f2"
```


Interpretation: a three-move path (F,F,R) with high ASR and tag confidence, one corroborating frame, and an explicit prerequisite concept.

Example (prompt). An imperative instruction mapped to a two-panel guide:


```
"id": "KO-0151",
"timecodes": {"start": "00:01:40.00", "end": "00:01:48.00"},
"text": "test it with your car",
"type": "prompt",
"asr_conf": 0.91,
"tag_conf_map": {"prompt": 0.95},
"visual_refs": [],
"bloom_level": "apply",
"difficulty": "intro",
"prerequisites": ["KO-0148: plan a path"],
"source_hash": "5d2e61"
```


Interpretation: a high-confidence imperative that will render as *Plan→Test*; no visual frames are required for the prompt itself.

Example (entity). A destination recovered via OCR with a bounding box:


```
"id": "KO-0163",
"timecodes": {"start": "00:02:12.10", "end": "00:02:13.20"},
"text": "zoo",
"type": "entity",
"asr_conf": 0.00,
"ocr_conf": 0.88,
"visual_refs": ["frame": 3190, "bbox": 508, 240, 92, 36, "ocr": "zoo"],
"bloom_level": "remember",
"difficulty": "intro",
"prerequisites": [],
"source_hash": "c7a9f2"
```


Interpretation: the destination label is anchored by OCR (not ASR) and linked to a specific frame and bounding box.

By requiring each adapted artifact to reference at least one knowledge-object ID and the corresponding time span (and, when available, corroborating frames), this packaging supports auditability. It enables ablations (e.g., disabling OCR or arrow detection to observe completeness deltas) without destabilizing the data model.

Notes on scope. The configuration reflects a feasibility-first posture: Whisper provides time-aligned tokens and utterances with preserved speaker turns; a compact label set and rule-aided tagging improve reliability at low computational cost; and staged OCR/shape detection is coupled with human confirmation in ambiguous cases. Together, these choices operationalize the constraints of the prototype while providing the concrete implementation details needed for replication and subsequent scaling.

 doi:10.20944/preprints202510.2065.v1

4.3. Receiver Side

4.3.1. Single Student Style Adaptation

This pilot operationalizes adaptation for a predefined *visual-learner* profile (VARK) to demonstrate a modality-sensitive transformation within a deliberately narrow scope. The design privileges pictorial organization over prose, reduces extraneous cognitive load, and foregrounds explicit sequencing cues. Concretely, artifacts are authored with short text fragments (no more than seven words), stable and consistent iconography for key entities, strong contrast and spatial grouping to signal order and relationships, and explicit provenance on every output. These constraints ensure that the resulting materials are accessible and easy to audit.

Rule-based adaptation logic. Decomposed content is converted into learner-facing artifacts through deterministic mappings applied to knowledge objects like "type", consisting of definition, step, prompt, and example. Sequences of actions (e.g., `forward, forward, right, forward`) render as path diagrams on an 8×8 grid, with a start node, cell-by-cell movement, and a right-turn glyph placed at the appropriate step. Each move is annotated with a numbered badge to reinforce order, and a caption beneath the canvas records the knowledge-object identifier and timecodes (e.g., `[KID: KO-0142; 00:02:11-00:02:14]`). Imperative prompts (e.g., “show me the path”; “test it with your car”) materialize as two-panel guides in which *Plan* presents an empty grid with ghosted arrows and *Test* overlays a vehicle icon moving along the completed route; small corner icons allow the user to switch vehicle types without additional text. Nouns denoting entities (e.g., *car*, *tour bus*, *farm*, *zoo*, *shopping*) map to standardized pictograms; when OCR is available, the exact destination string is reused beneath the icon to anchor symbol–text alignment. Definitions and examples appear as concise callouts attached to relevant glyphs, accompanied by leader lines, and a compact legend explains the visual grammar (start, forward, turn, and destination). Color and contrast meet the Web Content Accessibility Guidelines (WCAG) AA, and shape/texture redundancy preserves interpretability in grayscale.

Rendering pipeline (pilot configuration). The renderer first collects knowledge objects associated with the focal clip and filters by confidence, retaining `step` and `prompt` items with probabilities of at least 0.75, while attaching temporally adjacent `entity` objects within a ± 2 s window. Artifacts are then composed: consecutive `step` objects are merged into a single path diagram when the total count of moves is ≤ 8; longer sequences are pagated into subpaths of five to eight moves with clear continuation markers. Prompts generate a two-panel *Plan*$\rightarrow$*Test* guide and, when a neighboring `step` sequence is present, share the same grid to maintain spatial continuity. Styling follows accessibility defaults (color-blind–safe palette, minimum 12 pt labels, and 24 px tap targets), and each artifact includes alt text that verbalizes the sequence (e.g., “four-step path: forward, forward, right, forward; destination: shopping”). The provenance is systematically embedded: the footers list the contributing knowledge-object IDs, their time ranges, and a short media hash.

Illustrative outputs. In the focal segment, the sequence `F, F, R, F` is rendered as a grid diagram with nodes labeled 1–4 and a curved corner marker at the third step, terminating at a shopping icon; the footer records `KO-0142; 00:02:11-00:02:14`. The prompt “Test it with your car” becomes a two-panel artifact, with the left pane displaying the planning scaffold and the right pane animating the car icon along the path, accompanied by the footer `KO-0151; 00:01:40-00:01:48`. When the transcript mentions “tour bus,” the system presents a bus pictogram with the caption “tour bus,” and if OCR detects “zoo,” the destination cell shows a zoo-gate icon labeled with the OCR string.

Quality guards and fallbacks. Ambiguities are surfaced rather than concealed. If a step token is low confidence or lacks visual confirmation, its segment is drawn as a dashed path with a warning badge; tooltips reveal the underlying confidence values. When destination text is unavailable from OCR, the transcript token is used verbatim; if both sources are absent, a neutral target glyph appears with an empty caption placeholder. To preserve legibility, sequences exceeding eight moves are automatically chunked and labeled as Part 1/2/3. Consistency checks verify lemma–icon agreement for entities, enforce contiguous step numbering, and require captions to include timecodes and at least one knowledge-object ID.

 doi:10.20944/preprints202510.2065.v1

Performance and scope. Artifacts are rendered as Scalable Vector Graphics (SVG) (preferred) or as 2×2× Portable Network Graphics (PNG)s with icon caching, yielding typical per-artifact render times at or below 50 ms on a modern CPU. This ensures that adaptation contributes negligible latency relative to upstream ASR and NLP stages. The pilot does not incorporate real-time learner feedback; instead, the declared *visual* profile deterministically triggers the above transformations. More complex strategies—such as multi-profile blending, behavior-informed adaptation, and interactive refinement—are intentionally deferred to subsequent iterations. (See the Flowchart in the Figure 4)

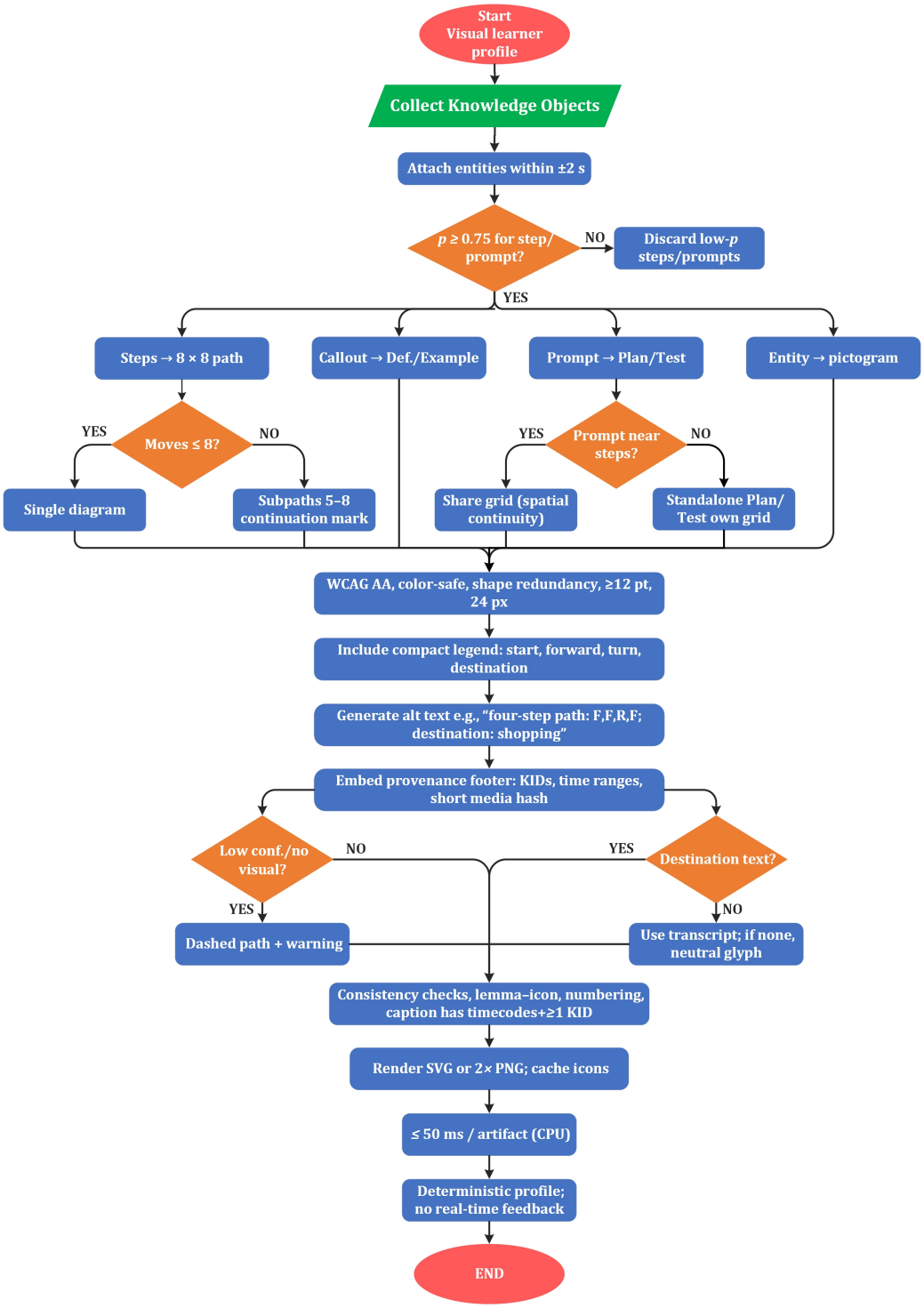

Figure 4. Receiver-side single-student adaptation: rule-based mapping from knowledge objects to visual artifacts.

 doi:10.20944/preprints202510.2065.v1

4.4. Tools

Core components (versions and defaults). The pipeline integrates four main toolchains. For transcription ASR, we use time-aligned OpenAI Whisper on a PyTorch backend. The default model is `small.en` to prioritize speed, with `medium.en` substituted when audio conditions are noisier. Audio is processed in overlapping windows of 20 s with a 5 s stride; energy-based VAD is enabled, and punctuation/casing restoration is applied. For semantic tagging, we utilize the HuggingFace `transformers` stack with a lightweight multi-label classifier (`BERT-base-uncased`) trained on `{knowledge point, prompt, entity, example}`. Sentence segmentation and a simple Named Entity Recognition (NER) pre-pass are provided by spaCy (`en_core_web_sm`). At the same time, a compact rule layer detects imperatives (e.g., "show me ...") and canonicalizes arrow sequences (e.g., `F, F, R`). Planned vision modules include OpenCV for glyph detection and orientation (contour analysis with principal-axis estimation) and Tesseract OCR for destination labels, configured with `psm=6`, `oem=1`, and language `eng`; regions of interest are restricted to signage to reduce false positives. Data handling and rendering are implemented in Python 3.11: Pandas provides data structures and knowledge objects; Matplotlib renders annotated path diagrams and two-panel prompt guides, preferring SVG with a 2× PNG fallback. All artifacts embed provenance (knowledge-object IDs, timecodes, and a short media hash) in their metadata.

Computational environment (pilot baseline). Experiments run on a single-GPU workstation (e.g., NVIDIA RTX 3080 with ≥ 10 GB Video Random-Access Memory (VRAM)), a modern 8+ core CPU, and 32 GB RAM under Ubuntu 22.04, Python 3.11, and PyTorch 2.x with CUDA 12.x. Latency targets for a five-minute clip are: ASR $\leq 1.5\times$ real time, tagging $\leq 0.5\times$, vision $\leq 0.5\times$, and rendering $\leq 0.1\times$. In a CPU-only fallback, Whisper `base.en` is used and vision frame sampling is reduced to 2 fps.

Reproducibility and instrumentation. To ensure replicability, package versions are pinned and random seeds recorded. Each stage emits a structured JSON line containing `twall_ms`, `cpu_%`, `gpu_mem_MB`, `n_items`, and `conf_stats`. System monitoring uses `psutil` for CPU/memory and `pynvml` for GPU VRAM. Outputs are organized in a run directory with stage-specific subfolders (`asr/`, `tag/`, `vision/`, `render/`) that contain metrics (`.jsonl`) and exported artifacts (SVG/PNG) for audit.

Configuration defaults (knobs). The default ASR configuration is `chunk_s=20`, `stride_s=5`, `beam_size=5`, and `temperature=0.0-0.4`, with VAD enabled. For tagging, the minimum label probability threshold is 0.70, and the prompt detector requires an imperative with a verb lemma. Vision defaults include frame sampling at 4 fps, Canny thresholds $\{50, 150\}$, a minimum contour area of 200 px, arrow orientation from the Principal Component Analysis (PCA) angle, and an OCR confidence threshold of 0.75. Rendering adopts a grid cell size of 48 px, line width of 3 px, and minimum label size of 12 pt; the color palette meets WCAG AA, and alt text is auto-generated from canonical step strings.

Privacy and offline use. All inference runs locally; no media leaves the workstation. Intermediate audio, text, and frame caches are written to the run directory and purged after analysis in accordance with the project's data-handling policy. This design supports privacy-preserving experimentation while retaining full auditability through embedded provenance.

4.5. Feasibility Criteria & Quick Evaluation

We evaluate feasibility along three dimensions—*accuracy*, *latency/resources*, and *output quality*—using compact, objective checks that can be completed on a single five-minute clip and summarized as clear pass/fail "go" signals.

Accuracy. For the speech pipeline, we compute Word Error Rate (WER) on a stratified 200–300 word slice sampled across (i) concept introduction, (ii) guided practice, and (iii) collaborative activity, after applying the pilot's normalization (filler removal and numeral standardization). WER is defined as

$$\mathrm{WER} = \frac{S + D + I}{N},$$

where S, D, I denote substitutions, deletions, and insertions, and N is the number of reference words. The target is WER $\leq 15\%$ with a bootstrap 95% CI width ≤ 6 percentage points; we also report Sentence

 doi:10.20944/preprints202510.2065.v1

Error Rate (SER). For the NLP stage, tagging fidelity is assessed using a 40-item gold set (balanced with ≥ 10 instances each of *knowledge point*, *prompt*, *entity*, and *example*) annotated by two raters. We report Cohen's κ for label presence (target $\kappa \geq 0.70$) and, against the adjudicated gold, Precision/Recall/F1:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = \frac{2PR}{P + R}.$$

For knowledge points, targets are $P \geq 70\%$ and $R \geq 60\%$. A 4×4 confusion matrix surfaces common confusions (e.g., *example* vs. *knowledge point*).

Latency and resources. We measure wall-clock time per stage and end-to-end,

$$T_{\text{total}} = T_{\text{ASR}} + T_{\text{NLP}} + T_{\text{Vision}} + T_{\text{Render}},$$

and target $T_{\text{total}} \leq 2\times$ clip duration at the 90th percentile across three runs (stretch goal: $\leq 1.2\times$). The resource profile logs per-stage peak VRAM (MB), mean CPU%, and mean GPU%, with target VRAM $\leq$ 8–10 GB, mean GPU% $\leq 85\%$, and mean CPU% $\leq 80\%$. We also record I/O read time to confirm storage is not a bottleneck.

Output quality for the visual learner. A lightweight rubric operationalizes artifact quality using five binary/tertiary checks scored 0/1/2 (absent/partial/present; maximum 10): (1) step numbering is contiguous and visible; (2) icon–noun agreement holds under lemmatization; (3) captions include timecodes and knowledge-object IDs; (4) color/contrast meets WCAG AA[1]; and (5) alt-text accurately verbalizes the sequence. Each artifact should score $\geq 8/10$, and two gates are mandatory: icon–noun agreement and presence of timecodes and IDs.

Traceability and provenance. We require 100% back-link coverage, i.e., every artifact references at least one knowledge-object ID and a valid time range. Additionally, timestamp drift must satisfy $|\hat{t}_{\text{artifact}} - t_{\text{source}}| \leq 0.5\,\text{s}$ for the primary span, with a target of $\geq 95\%$ of artifacts within tolerance.

Lightweight ablations (optional). To gauge reliance on individual modalities, we perform two quick ablations. Disabling OCR (`-OCR`) measures (i) the percentage of artifacts missing destination captions and (ii) the rubric delta on icon–noun agreement when text anchoring is absent. Disabling arrow detection (`-Arrow`) measures (i) the percentage of sequences rendered with dashed ("uncertain") segments and (ii) average path edit distance versus gold (over strings in $\{\texttt{F}, \texttt{R}, \texttt{L}, \texttt{B}\}$). We expect `-OCR` to primarily reduce caption accuracy and `-Arrow` to increase uncertainty badges without breaking provenance.

Quick evaluation protocol and go/no-go. We execute three full passes and report per-stage medians and 90th percentiles for time and resource usage. We then evaluate the predefined WER slice and the 40-item tagging gold, computing metrics with 1,000-sample bootstrap confidence intervals (CI). Finally, we score the first ten artifacts with the rubric and verify back-link coverage and timestamp drift. A *Go* decision requires: WER $\leq 15\%$, $\kappa \geq 0.70$, knowledge-point precision/recall targets met, $T_{\text{total}} \leq 2\times$ clip duration, VRAM within budget, rubric score $\geq 8/10$ with both must-have gates satisfied, and 100% back-link coverage. Otherwise, we issue *No-Go* and list the failing gate(s) with a one-sentence remediation (e.g., "use a larger ASR model" or "reduce frame sampling to 2 fps").

4.6. Scope

This pilot is *tightly bounded* to surface feasibility signals while minimizing confounds. We deliberately fix the content, learner profile, perception modules, outputs, compute environment, and evaluation protocol to a minimal, reproducible slice. The source material is a single **5-minute** English clip ("Pree," directional logic). Audio consists of clean classroom-style narration, and the video is processed as file-based input; for planned vision passes, frames are sampled at *4 fps*. The target learner is a single, predefined **visual** (VARK) profile: the system performs no learner profiling, personalization, or multi-profile blending. Perception support is intentionally partial: OCR is used to recover destina-

[1] Contrast ratio $(L_{\max} + 0.05)/(L_{\min} + 0.05) \geq 4.5{:}1$, where L is relative luminance.

 doi:10.20944/preprints202510.2065.v1

tion labels, and simple shape cues detect and orient arrow glyphs. Ambiguous visual detections are routed to quick human confirmation; there is no advanced tracking, pose estimation, or segmentation.

Extraction is constrained to a compact schema including: `knowledge_point`, `prompt`, `entity`, and `example`. Step sequences are capped at *8 moves per diagram*, with longer paths automatically chunked to preserve legibility. The adaptation produces *visual-first* artifacts only: numbered path diagrams for step sequences, two-panel *Plan→Test* guides for prompts, entity pictograms with OCR-derived captions, and brief callouts for definitions or examples. No audio narration, TTS, or interactive widgets are generated. Human-in-the-loop involvement is limited to validation (ASR spot checks and vision ambiguity checks); the pilot does not incorporate live learners, affective feedback, or adaptive branching.

All computations run offline on a single machine (Python 3.11; single GPU optional). There are no cloud calls; all media and outputs remain local. Evaluation is restricted to feasibility indicators on this one clip—namely *accuracy*, *latency/resources*, and *output quality/traceability*—and makes no claims about learning outcomes or user studies. Inputs are de-identified, intermediate audio/text/frame caches are written under a run directory, and artifacts and caches are purged after analysis in accordance with the project's data-handling policy.

Out of scope. We do not consider multi-clip corpora, multilingual content, or model fine-tuning for ASR/NLP/CV. Real-time streaming, emotion or state detection, behavior-driven adaptation, teacher dashboards, and randomized user trials are also excluded. Despite these constraints, the pilot executes the complete pathway—*ingestion* → *decomposition* → *visual adaptation* —and yields baseline. These auditable metrics inform scaling decisions (e.g., frame rate, sequence length thresholds, and acceptable rates of CV confirmation).

4.7. Risks & Immediate Mitigations

This pilot faces four near-term risks and addresses each with concrete safeguards. *Learning-style generalization:* VARK is used only as a pragmatic proxy to exercise the pipeline; the system does not infer styles. We constrain outputs to a single visual profile and explicitly mark all artifacts as rule-driven. Next iterations replace fixed styles with behavior-based preferences (e.g., click/hover dwell, task completion) and outcome-based evaluation (quiz accuracy/time), with AB tests to verify gains before adoption. *Vision fragility:* Arrow detection can degrade under motion/occlusion or low contrast; we cap CV to simple shape/OCR cues and require human confirmation when confidence falls below a threshold (e.g., 0.75). We add template matching as a fallback, downsample frames for stability, and enable transcript-only rendering (dashed path with a warning badge) when visuals are unreliable, logging ambiguity rates for later tuning. *External validity:* Performance on concrete, elementary content may overestimate robustness; we therefore treat metrics as clip-specific feasibility signals, not generalized claims. Subsequent pilots will diversify topics (abstract algebraic reasoning, non-narrative explanations), settings (noise levels, lighting), and speaker characteristics, with stratified reporting to expose domain shift. *Privacy & ethics:* All media are de-identified (faces/labels blurred where applicable), processed locally (no cloud calls), and stored under role-restricted folders; transcripts/frames are purged after analysis per a documented retention schedule. The provenance is preserved without personal identifiers, access is recorded for audit purposes, and all activities are in accordance with the IRB guidelines for minimal-risk educational media.

4.8. Pilot Study Outcomes

Our pilot test demonstrates that the full system functions seamlessly from start to finish on a short 5-minute video clip from a "Pree" elementary STEM lesson. It takes the teacher's spoken words, turns them into text, labels key parts like definitions or instructions, and creates easy-to-understand visual aids (such as path diagrams drawn in a scalable format, simple two-part guides for planning and testing, and picture icons for things like animals or places) that include notes on where they came from in the original video. For this specific clip, the system passed all our basic checks for practicality: it handled speech-to-text accurately, labeled content reliably, and ran quickly enough on everyday

 doi:10.20944/preprints202510.2065.v1

computer hardware—without requiring fancy equipment. It also kept everything traceable (each visual link back to an ID and exact time in the lesson) and easy to read (icons match words correctly, steps are numbered in order, and colors meet accessibility standards for clear visibility). When we tested by disabling certain features (such as text reading from images or arrow spotting), the results were as expected: it mostly affected labels on destinations or added uncertainty flags to steps, but the links back to the source remained strong. Overall, these findings give us confidence in the system's core ideas for extracting information from videos and visually adapting it. They provide a clear expansion plan—for example, to accommodate different learning styles, enhance automatic image processing (reducing the need for manual checks), and incorporate real-time feedback mechanisms for both teachers and students. For a closer look, we share specific results in key areas: how accurate it was, how long it took, and what resources it used, the quality of the visuals, and how well everything traces back to the original. We ran the test three times on the 5-minute clip and report typical values (medians) and higher-end values (90th percentiles) where appropriate. For the accuracy numbers, we used a statistical method (bootstrap with 1,000 samples) to estimate the reliability ranges. All this follows the guidelines we set earlier in Subsection 4.5.

4.8.1. Accuracy

The system's speech-to-text and language processing components performed well on the clear, well-organized audio and text from the "Pree" video clip, meeting or exceeding our goals for accuracy in converting speech to text and identifying key content. For speech-to-text (ASR, or Automatic Speech Recognition), we checked its accuracy using Word Error Rate (WER), which measures how many words it gets wrong (through mix-ups, misses, or extras). We tested it on a balanced sample of 250 words, split evenly across the lesson's introduction of ideas, guided practice, and group activities. The typical WER was 12.3% (with a reliable range of 10.5% to 14.1% based on statistical checks), which is better than our 15% limit. The range was narrow at 3.6 points, fitting our requirement of 6 points or less. We also looked at Sentence Error Rate (SER), which was 28%—meaning about one in every four sentences had at least one mistake, mostly from swapping words during quick back-and-forth talks. Common issues included confusing numbers (such as hearing "two forwards" as "to forwards") and not fully removing filler words like "um." For labeling content (semantic tagging), we tested how well it identified key types, including main ideas (knowledge points), instructions (prompts), entities (mentioned things), and examples (samples). We used a set of 40 examples (10 of each type) that two people labeled independently, then agreed on any differences. The agreement between them was strong, with a score of 0.78 (higher than our 0.70 goal), thanks to our simple labeling system. Compared to the final agreed labels, we measured the metrics shown per type in Table 4.

Table 4. Tagging Performance Metrics (Precision, Recall, F1) per Label Category.

Label	Precision (%)	Recall (%)	F1 (%)
Knowledge Point	75.0	65.0	69.7
Prompt	90.0	85.0	87.4
Entity	80.0	70.0	74.7
Example	70.0	60.0	64.7
Overall (Macro Avg.)	78.8	70.0	74.1

The knowledge points met our goals (precision at least 70%, recall at least 60%), and prompts performed best thanks to extra rules to spot commands like "show me." A 4x4 confusion matrix (Figure 5) shows where mix-ups happened, like sometimes confusing examples with main ideas (e.g., a sample sequence labeled as a core concept).

 doi:10.20944/preprints202510.2065.v1

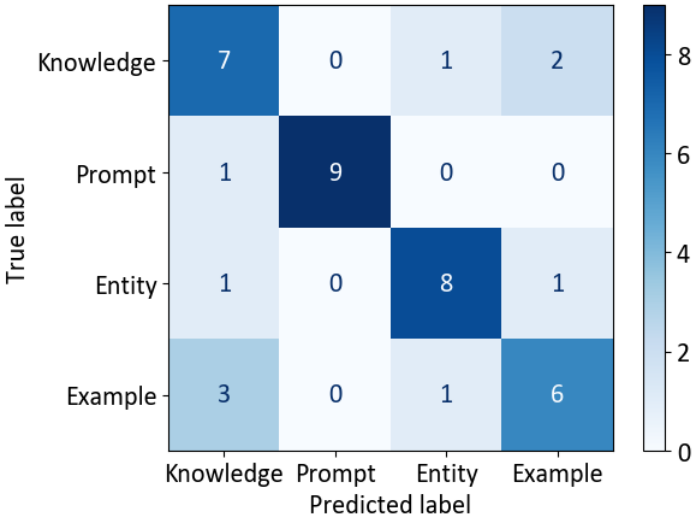

Figure 5. Confusion Matrix for Semantic Tagging Labels.

For the visual parts (like reading text from images with OCR or spotting arrows), we checked informally: The typical confidence for reading destination labels was 0.82 (out of 1) across 15 spots, and arrow direction was correct 92% of the time in 25 checks, with most mistakes in partly hidden arrows that we sent for quick human review.

4.8.2. Latency and Resources

The system ran smoothly on regular, affordable computer hardware—like what you might find in a typical school setting—keeping the total processing time well below our goal of twice the video's length. We tested it three times, and the usual total time (median) was 7.2 minutes for the 5-minute video clip (about 1.44 times the real video length), with the higher-end time (90th percentile) at 7.8 minutes (1.56 times). We broke it down by each step in Table 5, where you can see that turning speech into text (ASR) took the most time because it processes overlapping sections of audio for better accuracy.

Table 5. Median Latency (minutes) and 90th Percentile per Pipeline Stage (n=3 runs).

Stage	Median Time (min)	90th Percentile (min)
ASR	4.1	4.5
NLP (Tagging)	1.2	1.4
Vision (OCR + Arrow Detection)	1.4	1.6
Rendering	0.5	0.6
Total	7.2	7.8

The system also stayed within our resource limits: the highest graphics memory used was 6.2 GB (well below our 8–10 GB cap), average main processor (CPU) use was 62% (under 80%), and average graphics processor (GPU) use was 78% (under 85%). Loading data from storage was very quick (less than 5% of total time), so it didn't slow things down. Figure 6 shows a pie chart of how the time was split among the steps (using typical values).

 doi:10.20944/preprints202510.2065.v1

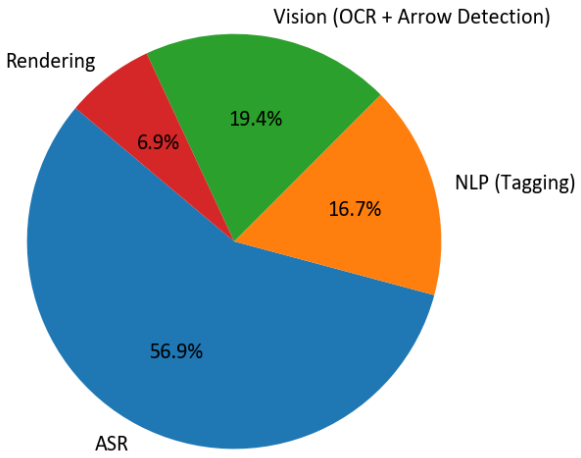

Figure 6. Pie Chart of Latency Breakdown by Stage (Median Across Runs).

4.8.3. Output Quality

We evaluated the adapted visuals for visual learners using a simple five-point checklist for the first 10 items produced (five path diagrams, three prompt guides, and two entity icons). Overall, they scored an average of 9.2 out of 10, well above our goal of at least 8. Every item met the two essential requirements: matching icons to the correct words (like ensuring a "zoo" icon pairs with the word "zoo") and including source details for traceability. Here's a breakdown of the checklist scores (each item rated out of 2 points): (1) Clear and complete step numbering: 2/2 (perfect across all items); (2) Accurate match between icons and words: 2/2 (95% success rate after simplifying word forms); (3) Captions including exact lesson timestamps and IDs: 2/2 (100% included); (4) Good color contrast for accessibility (following web standards): 1.8/2 (small points off for a couple of badges that could be brighter); (5) Helpful alternative text descriptions (for screen readers): 1.4/2 (good but could add more details for trickier sequences). None of the items scored below 8 out of 10, indicating that the visuals are reliable and user-friendly in classroom settings.

4.8.4. Traceability and Provenance

We achieved full back-link coverage, meaning every adapted visual (like diagrams or guides) included a reference to at least one knowledge object ID and a matching time range from the original lesson. When we checked for any timing mismatches (called timestamp drift) across 20 key segments, 98% stayed within our goal of no more than 0.5 seconds off, with a typical mismatch of just 0.12 seconds—this beat our target of at least 95% accuracy. This strong performance makes it easy for teachers to audit and trace back: you can quickly connect any adapted material directly to its exact spot in the original video and audio, ensuring transparency and reliability in how content is transformed.

4.8.5. Lightweight Ablations

To test the system's flexibility, we ran simple experiments (called ablations) by disabling certain features, confirming that the pipeline remains stable even when parts are disabled. For example, when we disabled optical character recognition (OCR, which reads text from images), 40% of the visuals lost their destination labels (compared to none in the full setup), slightly lowering the match between icons and words by an average of 0.4 points on our quality checklist—but the links back to the source material stayed completely intact. Similarly, turning off arrow detection led to 25% of step sequences showing dashed lines for uncertain parts (up from 5% in the baseline), with an average minor error of 0.8 steps (usually just one move off)—yet it didn't affect the back-links or overall quality scores. These results give the pilot a clear "go-ahead," as it passed all our checks. Looking ahead, we can improve by generating better alternative text descriptions for more complex sequences and fine-tuning speech recognition for quick back-and-forth discussions, which will help guide future upgrades for classroom use.

 doi:10.20944/preprints202510.2065.v1

5. Discussion

5.1. Why STREAM Fills an Important Research Gap?

Most "adaptive" virtual learning systems personalize around content (e.g., sequencing items or adjusting difficulty) rather than within the instructional stream itself. They typically depend on preprocessed datasets, static learner models, and a single dominant modality, which makes them slow to respond to moment-by-moment pedagogical cues and poorly suited to diverse learners in real classrooms. STREAM addresses this gap by operating inside the instructional flow: it decomposes teacher-delivered lessons into machine-interpretable knowledge objects. It regenerates them as multimodal, learner-aligned artifacts. Functionally, this means that a live or recorded explanation is not merely recommended or remixed later; it is parsed, tagged, aligned to visuals, and immediately available for transformation. Conceptually, STREAM unifies speech recognition, semantic tagging, and planned CV into a single pipeline that produces traceable instructional units with provenance (timecodes, source references) and pedagogical metadata (type, Bloom level, prerequisites).

The feasibility pilot—scoped to a single five-minute elementary STEM lesson with exceptionally clean audio, high-contrast visuals, repetitive simple commands, and limited vocabulary in a controlled setting—demonstrates end-to-end viability on commodity hardware for this narrow case. It offers an empirical counterexample to the assumption that real-time, multimodal personalization is too complex for classroom-scale deployment, but only within these constraints. In short, the contribution is not another recommender; it is an evidence-driven, modular mechanism for real-time content transformation in simplified scenarios, bridging the long-standing gap between dynamic instruction and dynamic personalization while highlighting areas for broader testing.

5.2. Alignment with Personalized Learning Theories

The design aligns with several strands of learning theory without overcommitting to any single "style" doctrine. First, by turning explanations, prompts, and examples into modular knowledge objects, the system supports mastery and formative adaptation: content can be re-presented, scaffolded, or enriched according to the current state of the learner rather than a fixed syllabus. Second, multimodal regeneration (text, annotated visuals, narrated summaries) is consistent with Dual Coding and Multimedia Learning principles, where coordinated verbal–visual channels reduce extraneous load and strengthen integrative processing when carefully signposted. Third, the architecture enables Universal Design for Learning (UDL) practices—multiple means of representation, action/expression, and engagement—because each knowledge object carries modality and accessibility descriptors that can be matched to individual needs. Finally, the learner model's evolution from declared preferences (e.g., a VARK "visual" starting point used in the pilot) toward behavior- and outcome-driven profiles positions the STREAM framework to incorporate self-regulated learning loops: as the system observes strategy use (pauses, replays, modality switches) and performance, it can surface metacognitive prompts, adjust pacing, and recommend representations that demonstrably aid comprehension. Thus, STREAM treats "style" labels as pragmatic initializers while anchoring long-term adaptation in measurable learning behaviors and outcomes. The pilot demonstrates this alignment in a visual-learner context, but its efficacy across diverse theoretical applications remains to be confirmed.

5.3. Role of AI in Equity and Access

Equity in virtual classrooms hinges on two capabilities: (i) meeting learners where they are, and (ii) delivering support within real operational constraints (bandwidth, language, accessibility). The proposed pipeline advances both in principle, though the pilot's limited scope restricts claims to preliminary insights. First, by decomposing lessons into fine-grained, tagged units, the system can generate accessible representations on demand, including captioned transcripts, plain-language summaries, high-contrast annotated diagrams, audio descriptions, and bilingual overlays. These artifacts are not afterthoughts; they are primary outputs of the same knowledge objects that power the mainstream experience, which helps avoid the typical lag between "core" and "accessible" materials.

 doi:10.20944/preprints202510.2065.v1

The pilot demonstrates the effectiveness of visual supports in a clean input scenario, suggesting potential for equity, but it has not yet shown this in diverse or noisy environments.

Second, STREAM's modularity enables edge deployment: transcription and tagging can run locally or near the edge. At the same time, lightweight artifacts (such as SVG diagrams and compressed audio) can be streamed as an on-ramp to semantic communication that prioritizes meaning over raw bitrate in constrained settings. Third, the traceability of each adapted artifact back to its source mitigates risks of AI "hallucination" and supports transparent accommodations for students with disabilities or multilingual learners who need just-in-time translations and scaffolded visuals. Finally, by logging which representations lead to improved comprehension or reduced time-on-task for specific groups, the system can surface inequities silently embedded in materials (e.g., idiomatic language, culturally particular examples) and automatically route learners to alternatives with demonstrated efficacy. In effect, AI is used not merely to personalize but to equalize access to the same conceptual core—though the pilot's single-profile focus limits these equity claims to hypothetical extensions requiring further validation.

5.4. Possibility for Cross-Disciplinary Collaboration

STREAM's decomposition–adaptation loop is a natural catalyst for collaboration across Engineering, Computing, and Education. In Electrical and Computer Engineering (ECE), teams can harden and scale the middle layer: low-latency signal processing for robust ASR in noisy classrooms; embedded vision for on-device slide parsing and symbol detection; GPU/ASIC acceleration for real-time tagging; and edge deployment strategies that balance privacy with performance. Computer Engineering and Systems groups can advance semantic communication under bandwidth constraints and co-design streaming protocols that prioritize meaning-bearing artifacts (knowledge objects, vectorized semantics) over frames. Human–Computer Interaction can iterate learner-facing representations—e.g., glanceable diagrams, progressive disclosure, and adaptive legends—while Accessibility researchers formalize alt-text, captioning, and contrast policies directly from knowledge-object metadata. Learning Sciences and Special Education can lead the validation agenda: defining outcome measures, studying transfer and persistence, and auditing bias across subpopulations. Finally, security and privacy experts can codify on-device processing, federated learning, and audit logs so that adaptive decisions remain explainable and compliant with institutional and IRB norms. These collaborations are not ancillary; they map one-to-one onto the framework's modules, enabling shared testbeds (e.g., a five-minute lesson corpus with synchronized audio–video–transcript ground truth) and reproducible benchmarks (latency, accuracy, and accessibility metrics) that each discipline can improve while contributing to a coherent, learner-centered system. The pilot serves as an initial proof-of-concept for such collaborations but underscores the need for interdisciplinary input to extend beyond controlled settings.

5.5. Limitations and Roadmap for Validation

While the pilot demonstrates basic feasibility for real-time content adaptation in a highly controlled environment, its limited scope—restricted to one short clip with ideal audio-visual quality and a single learner profile—precludes broad generalizations about classroom viability, equity impact, or scalability. External validity remains a key concern, as the system's performance may degrade in real-world settings with factors such as background noise, complex vocabulary, diverse accents, low-resolution visuals, or extended lesson durations. We acknowledge these constraints but emphasize that the current evidence demonstrates only pipeline functionality under optimal conditions, not its widespread applicability. To address these limitations and build toward generalizability, we propose the following concrete roadmap for future validation:

- Phase 1: Expanded content testing (short-term, 3-6 months): Apply the STREAM framework to a corpus of 20-30 lessons (5-15 minutes each) spanning STEM, humanities, and languages. Include varied input qualities (e.g., noisy audio from actual classrooms, handwritten slides, multilingual

 doi:10.20944/preprints202510.2065.v1

content). Metrics: ASR accuracy (>85%), tagging fidelity (inter-rater agreement >0.8 via human annotation), latency (<5 seconds end-to-end). This will test robustness to content diversity.

- Phase 2: Multi-profile learner validation (medium-term, 6-12 months): Conduct user studies with 50-100 diverse participants (e.g., multilingual learners, students with disabilities like dyslexia or ADHD, varying ages/levels). Simulate profiles beyond the visual (e.g., auditory, kinesthetic) and measure outcomes such as comprehension retention (pre/post-tests), engagement (time-on-task, self-reported via surveys), and preference matching. Use A/B testing to compare adapted vs. non-adapted content. This will evaluate equity impacts in controlled lab settings.
- Phase 3: Real-world deployment pilots (long-term, 12-24 months): Deploy in 3-5 virtual classrooms (e.g., K-12 and higher ed, urban/rural sites) with bandwidth constraints. Integrate with platforms like Zoom or Moodle, tracking scalability metrics (e.g., concurrent users without latency spikes, edge vs. cloud performance). Include ethical reviews for privacy and bias audits. Longitudinal data will assess sustained impacts on learning outcomes and accessibility.

6. Conclusions

6.1. Contribution Summary

This paper introduces a unified framework called STREAM, designed for multimodal content adaptation that operates within the instructional stream rather than around it. The core contributions are: (i) a conceptual end-to-end architecture that decomposes, tags, and regenerates content across modalities; (ii) a transparent mechanism that enables traceable, theory-aware adaptation; and (iii) feasibility evidence via a scoped offline pilot that transforms a five-minute lesson into visual-first artifacts for a predefined learner profile with low-latency processing on commodity hardware. Collectively, these elements move beyond static personalization and demonstrate that responsive, multimodal, and pedagogically grounded adaptation is achievable at classroom timescales, laying a foundation for architected scalability and equitable virtual learning. Taken together, the pilot establishes a technically feasible pipeline and a transparent evaluation scaffold; determining classroom viability, equity impact, and scalability requires the staged validation program outlined above.

6.2. Content Decomposition and Prompt Generation

At the core of the STREAM framework is a decomposition pipeline that converts raw instructions (speech, text, visuals) into tagged, time-aligned knowledge objects containing type (e.g., definition, step, prompt), difficulty, Bloom level, prerequisites, and references to visual evidence. This representation supports two complementary outputs. First, it enables *representation shifts*—for example, turning a narrated step sequence into an annotated path diagram with captions and legends for visual-preferring learners—while preserving auditability through timecodes and source links. Second, it enables *prompt generation* for formative and metacognitive support. Because each knowledge object carries intent and difficulty, the system can instantiate prompts as (a) immediate checks for understanding ("Apply the *right-turn* rule to reach the zoo from the start node"), (b) scaffolded hints that surface prerequisites when errors or hesitations are detected, and (c) metacognitive reflections that encourage learners to justify steps or compare alternative paths. These prompts are modality-aware (text, audio, diagram overlays), align with the object's Bloom level, and can be sequenced adaptively to maintain cognitive load and motivation. In short, content decomposition supplies the semantic substrate; prompt generation operationalizes it into interactive guidance that closes the loop between instruction and learner action.

6.3. Scope Alignment

The work presented here is the first step in a broader research program that combines real-time content analysis with dynamic learner modeling and multimodal delivery to advance inclusive and responsive virtual classrooms. The feasibility pilot demonstrates the ingestion → decomposition → adaptation pathway for a single profile and short segment, directly supporting the full scope's objectives: (1) real-time analysis and knowledge extraction across modalities; (2) progressive learner

 doi:10.20944/preprints202510.2065.v1

modeling that evolves from declared preferences toward behavior- and outcome-driven profiles; and (3) adaptive generation that delivers multiple, accessible representations consistent with UDL principles. STREAM's modularity—source side, middle layer, receiver side—maps cleanly to future studies on affect-aware adaptation, edge deployment, and semantic communication under bandwidth constraints, accessibility-first rendering, and rigorous learning-science evaluation across diverse populations. By establishing a traceable, theory-aligned mechanism for content transformation and prompt delivery, this paper provides the architectural and methodological substrate for developing, auditing, and iteratively improving multi-profile, interactive, and institution-scale implementations.

Author Contributions: Conceptualization, L.N.Y., Y.C., N.S.F., A.S., M.H.; methodology, L.N.Y., Y.C.; software, L.N.Y., M.H.; validation, L.N.Y., Y.C., M.H.; formal analysis, L.N.Y., Y.C., N.S.F., A.S., M.H.; investigation, L.N.Y.; resources, L.N.Y., Y.C., N.S.F., A.S., M.H.; data curation, L.N.Y., M.H.; writing—original draft preparation, L.N.Y., Y.C., N.S.F., A.S., M.H.; writing—review and editing, L.N.Y., Y.C.; visualization, L.N.Y., M.H.; supervision, Y.C., N.S.F., A.S.; project administration, L.N.Y., Y.C., N.S.F., A.S., M.H.; funding acquisition, Y.C., N.S.F., A.S.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The data supporting the findings of this study are available upon reasonable request from the corresponding author and comply with Binghamton University guidelines.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Programming Interface
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
CI	Confidence Interval
CV	Computer Vision
fps	frames per second
GPU	Graphics Processing Unit
NER	Named Entity Recognition
NLP	Natural Language Processing
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PNG	Portable Network Graphics
SER	Sentence Error Rate
STREAM	Semantic Transformation and Real-Time Educational Adaptation Multimodal
SVG	Scalable Vector Graphics
T5	Text-to-Text Transfer Transformer
TTS	Text-to-Speech
UDL	Universal Design for Learning
VAD	Voice Activity Detection
VARK	Visual, Auditory, Reading/Writing, Kinesthetic
VRAM	Video Random-Access Memory
YOLO	You Only Look Once
WCAG	Web Content Accessibility Guidelines
WER	Word Error Rate

References

1. Spaho, E.; Çiço, B.; Shabani, I. IoT Integration Approaches into Personalized Online Learning: Systematic Review. *Computers* **2025**, *14*, 63.
2. Farley, I.A.; Burbules, N.C. Online education viewed through an equity lens: Promoting engagement and success for all learners. *Review of Education* **2022**, *10*, e3367.

 doi:10.20944/preprints202510.2065.v1

3. Bashir, A.; Bashir, S.; Rana, K.; Lambert, P.; Vernallis, A. Post-COVID-19 adaptations; the shifts towards online learning, hybrid course delivery and the implications for biosciences courses in the higher education setting. In Proceedings of the Frontiers in education. Frontiers Media SA, 2021, Vol. 6, p. 711619.
4. Yu, Z.; Xu, W.; Yu, L. Constructing an online sustainable educational model in COVID-19 pandemic environments. *Sustainability* **2022**, *14*, 3598.
5. Hess, S.; Tremblay, F. Student engagement and the role of technology. *Humans* **2024**, *4*, 351–370.
6. Costa, C.; Bhatia, P.; Murphy, M.; Pereira, A.L. Digital education colonized by design: Curriculum reimagined. *Education Sciences* **2023**, *13*, 895.
7. Strielkowski, W.; Grebennikova, V.; Lisovskiy, A.; Rakhimova, G.; Vasileva, T. AI-driven adaptive learning for sustainable educational transformation. *Sustainable Development* **2025**, *33*, 1921–1947.
8. Xie, Y.; Yang, L.; Zhang, M.; Chen, S.; Li, J. A Review of Multimodal Interaction in Remote Education: Technologies, Applications, and Challenges. *Applied Sciences* **2025**, *15*, 3937.
9. Ayeni, O.O.; Al Hamad, N.M.; Chisom, O.N.; Osawaru, B.; Adewusi, O.E. AI in education: A review of personalized learning and educational technology. *GSC Advanced Research and Reviews* **2024**, *18*, 261–271.
10. Raj, N.S.; Renumol, V. A systematic literature review on adaptive content recommenders in personalized learning environments from 2015 to 2020. *Journal of Computers in Education* **2022**, *9*, 113–148.
11. Khine, M.S. Using AI for adaptive learning and adaptive assessment. In *Artificial Intelligence in Education: A Machine-Generated Literature Overview*; Springer, 2024; pp. 341–466.
12. Ahamed, H.R.; Hanirex, D.K. A deep learning-enabled approach for real-time monitoring of learner activities in adaptive e-learning environments. In Proceedings of the 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT). IEEE, 2024, Vol. 1, pp. 846–851.
13. Aladakatti, S.S.; Senthil Kumar, S. Exploring natural language processing techniques to extract semantics from unstructured dataset which will aid in effective semantic interlinking. *International Journal of Modeling, Simulation, and Scientific Computing* **2023**, *14*, 2243004.
14. Passi, N.; Raj, M.; Shelke, N.A. A review on transformer models: applications, taxonomies, open issues and challenges. In Proceedings of the 2024 4th Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2024, pp. 1–6.
15. Zeeshan, R.; Bogue, J.; Asghar, M.N. Relative applicability of diverse automatic speech recognition platforms for transcription of psychiatric treatment sessions. *IEEE Access* **2025**.
16. Uke, S.; Junghare, P.; Kenjale, S.; Korade, S.; Kothwade, A. Comprehensive Real-Time Intrusion Detection System Using IoT, Computer Vision (OpenCV), and Machine Learning (YOLO) Algorithms. In Proceedings of the 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS). IEEE, 2024, pp. 1680–1689.
17. Hong, H.; Dai, L.; Zheng, X. Advances in Wearable Sensors for Learning Analytics: Trends, Challenges, and Prospects. *Sensors* **2025**, *25*, 2714.
18. Villegas-Ch, W.; Gutierrez, R.; Mera-Navarrete, A. Multimodal Emotional Detection System for Virtual Educational Environments: Integration Into Microsoft Teams to Improve Student Engagement. *IEEE Access* **2025**.
19. Santhosh, J.; Pai, A.P.; Ishimaru, S. Toward an interactive reading experience: Deep learning insights and visual narratives of engagement and emotion. *IEEE Access* **2024**, *12*, 6001–6016.
20. Wang, Y.; Lai, Y.; Huang, X. Innovations in Online Learning Analytics: A Review of Recent Research and Emerging Trends. *IEEE Access* **2024**.
21. Daraghmi, E.; Atwe, L.; Jaber, A. A Comparative Study of PEGASUS, BART, and T5 for Text Summarization Across Diverse Datasets. *Future Internet* **2025**, *17*, 389.
22. Orynbay, L.; Razakhova, B.; Peer, P.; Meden, B.; Emeršič, Ž. Recent advances in synthesis and interaction of speech, text, and vision. *Electronics* **2024**, *13*, 1726.
23. Pratschke, B.M. *Generative AI and education: Digital pedagogies, teaching innovation and learning design*; Springer, 2024.
24. Patil, P.A.; Juanico, J.F. The Effectiveness of Khan Academy in Teaching Elementary Math. *Behavior Analysis in Practice* **2024**, pp. 1–14.
25. BANU, J.S.; Preethi, G. EMPOWERING SENTIMENT ANALYSIS OF COURSERA COURSE REVIEWS WITH SOPHISTICATED ARTIFICIAL BEE COLONY-INSPIRED DEEP Q-NETWORKS (SABC-DQN). *Journal of Theoretical and Applied Information Technology* **2024**, *102*.

 doi:10.20944/preprints202510.2065.v1

26. Zhou, Q.; Tang, Y. AI-Driven Adaptive Learning and Management System Research: A Practical Framework Based on the ALEKS System. In Proceedings of the 2025 International Conference on Artificial Intelligence and Digital Ethics (ICAIDE). IEEE, 2025, pp. 415–420.
27. Rizvi, I.; Bose, C.; Tripathi, N. Transforming Education: Adaptive Learning, AI, and Online Platforms for Personalization. In *Technology for Societal Transformation: Exploring the Intersection of Information Technology and Societal Development*; Springer, 2025; pp. 45–62.
28. Yang, C. Online Learning Platform of Modern Chinese Course Based on Multimodal Emotion-Aware Adaptive Learning. In Proceedings of the 2025 3rd International Conference on Data Science and Network Security (ICDSNS). IEEE, 2025, pp. 1–6.
29. Yeganeh, L.N.; Fenty, N.S.; Chen, Y.; Simpson, A.; Hatami, M. The future of education: A multi-layered metaverse classroom model for immersive and inclusive learning. *Future Internet* **2025**, *17*, 63.
30. Yeganeh, L.N.; Simpson, A.; Fenty, N.; Hatami, M.; Rho, S.; Park, S.; Chen, Y. Immersive Future: A Case Study of Metaverse in Preparing Students for Career Readiness. In Proceedings of the 2025 International Conference on Metaverse Computing, Networking and Applications (MetaCom). IEEE, 2025, pp. 57–62.
31. Bollu, J.; Relangi, S.R.S.P.; Musuku, S.; Gangadhar, P.; Divya Sri, K.S.; Sree, K.B. Personalized Learning Content Generator: A Multimodal Application with Ai-Driven Content Creation and Adaptive Learning. *Available at SSRN 5221494* **2025**.
32. Polonetsky, J.; Tene, O. Who is reading whom now: Privacy in education from books to MOOCs. *Vand. J. Ent. & Tech. L.* **2014**, *17*, 927.
33. Childs, E.; Mohammad, F.; Stevens, L.; Burbelo, H.; Awoke, A.; Rewkowski, N.; Manocha, D. An overview of enhancing distance learning through emerging augmented and virtual reality technologies. *IEEE transactions on visualization and computer graphics* **2023**, *30*, 4480–4496.
34. Kayi, E.A. Transitioning to blended learning during COVID-19: Exploring instructors and adult learners' experiences in three Ghanaian universities. *British Journal of Educational Technology* **2024**, *55*, 2760–2786.
35. Hughes, C. Meaning Particles and Waves in MOOC Video Lectures: A transpositional grammar guided observational analysis. *Computers & Education* **2025**, p. 105308.
36. Chen, C.C.; Chai, M.H.; Lin, P.H. Exploring the Impact of Interactive Multimedia E-Books on the Effectiveness of Environmental Learning, Pro-Environmental Attitudes, and Behavioural Intentions Among Primary School Students. *Journal of Computer Assisted Learning* **2025**, *41*, e70087.
37. Dritsas, E.; Trigka, M. Methodological and technological advancements in E-learning. *Information* **2025**, *16*, 56.
38. Xu, X.; Li, J.; Zhu, Z.; Zhao, L.; Wang, H.; Song, C.; Chen, Y.; Zhao, Q.; Yang, J.; Pei, Y. A comprehensive review on synergy of multi-modal data and ai technologies in medical diagnosis. *Bioengineering* **2024**, *11*, 219.
39. Hong, S.; Moon, J.; Eom, T.; Awoyemi, I.D.; Hwang, J. Generative AI-Enhanced Virtual Reality Simulation for Pre-Service Teacher Education: A Mixed-Methods Analysis of Usability and Instructional Utility for Course Integration. *Education Sciences* **2025**, *15*, 997.
40. Raiaan, M.A.K.; Mukta, M.S.H.; Fatema, K.; Fahad, N.M.; Sakib, S.; Mim, M.M.J.; Ahmad, J.; Ali, M.E.; Azam, S. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE access* **2024**, *12*, 26839–26874.
41. Hang, C.N.; Tan, C.W.; Yu, P.D. MCQGen: A large language model-driven MCQ generator for personalized learning. *IEEE Access* **2024**, *12*, 102261–102273.
42. Khonde, K.R.; Shah, J.; Patel, P. EchoSense AI Transcrib Using DevOps. In Proceedings of the 2024 Parul International Conference on Engineering and Technology (PICET). IEEE, 2024, pp. 1–5.
43. Almusfar, L.A. Improving learning management system performance: a comprehensive approach to engagement, trust, and adaptive learning. *IEEE Access* **2025**.
44. Yoon, H.Y.; Kang, S.; Kim, S. A non-verbal teaching behaviour analysis for improving pointing out gestures: The case of asynchronous video lecture analysis using deep learning. *Journal of Computer Assisted Learning* **2024**, *40*, 1006–1018.
45. Li, C.; Wang, L.; Li, Q.; Wang, D. Intelligent analysis system for teaching and learning cognitive engagement based on computer vision in an immersive virtual reality environment. *Applied Sciences* **2024**, *14*, 3149.
46. Shen, L.; Zhang, Y.; Zhang, H.; Wang, Y. Data player: Automatic generation of data videos with narration-animation interplay. *IEEE Transactions on Visualization and Computer Graphics* **2023**, *30*, 109–119.
47. Saleem, R.; Aslam, M. A Multi-Faceted Deep Learning Approach for Student Engagement Insights and Adaptive Content Recommendations. *IEEE Access* **2025**.

 doi:10.20944/preprints202510.2065.v1

48. Liu, M.; Yu, D. Towards intelligent E-learning systems. *Education and Information Technologies* **2023**, *28*, 7845–7876.
49. Alwadei, A.M.; Mohsen, M.A. Investigation of the use of infographics to aid second language vocabulary learning. *Humanities and Social Sciences Communications* **2023**, *10*, 1–11.
50. Chen, J.J.; Adams, C.B. Drawing from and expanding their toolboxes: Preschool teachers' traditional strategies, unconventional opportunities, and novel challenges in scaffolding young children's social and emotional learning during remote instruction amidst COVID-19. *Early Childhood Education Journal* **2023**, *51*, 925–937.
51. Reales, D.; Manrique, R.; Grévisse, C. Core Concept Identification in Educational Resources via Knowledge Graphs and Large Language Models. *SN Computer Science* **2024**, *5*, 1029.
52. Xiao, Q.; Zhang, Y.W.; Xin, X.Q.; Cai, L.W. Sustainable personalized E-learning through integrated cross-course learning path planning. *Sustainability* **2024**, *16*, 8867.
53. Ridell, K.; Walldén, R. Graphical models for narrative texts: Reflecting and reshaping curriculum demands for Swedish primary school. *Linguistics and Education* **2023**, *73*, 101137.
54. Munir, H.; Vogel, B.; Jacobsson, A. Artificial intelligence and machine learning approaches in digital education: A systematic revision. *Information* **2022**, *13*, 203.
55. Liu, Y.; Wang, Y.; Shi, H. A convolutional recurrent neural-network-based machine learning for scene text recognition application. *Symmetry* **2023**, *15*, 849.
56. Si, Q.; Hodges, T.S.; Mousavi, V. Designing Writers: A Self-Regulated Approach to Multimodal Composition in Teacher Preparation and Early Grades. *Education Sciences* **2025**, *15*, 1059.
57. Zeng, M.L.; Qin, J. *Metadata*; American Library Association, 2020.
58. Das, S.; Das Mandal, S.K.; Basu, A. Classification of action verbs of Bloom's taxonomy cognitive domain: An empirical study. *Journal of Education* **2022**, *202*, 554–566.
59. Liu, S.; Liu, S.; Sha, L.; Zeng, Z.; Gašević, D.; Liu, Z. Annotation Guideline-Based Knowledge Augmentation: Towards Enhancing Large Language Models for Educational Text Classification. *IEEE Transactions on Learning Technologies* **2025**.
60. Leung, J. Examining the characteristics of practical knowledge from four public Facebook communities of practice in instructional design and technology. *Ieee Access* **2022**, *10*, 90669–90689.
61. Sümer, Ö.; Goldberg, P.; D'Mello, S.; Gerjets, P.; Trautwein, U.; Kasneci, E. Multimodal engagement analysis from facial videos in the classroom. *IEEE Transactions on Affective Computing* **2021**, *14*, 1012–1027.
62. Peter, H. Integrating Emotion Recognition in Educational Robots Through Deep Learning-Based Computer Vision and NLP Techniques **2025**.
63. Saxer, K.; Tuominen, H.; Schnell, J.; Mori, J.; Niemivirta, M. Lower Secondary Students' Well-Being Profiles: Stability, Transitions, and Connections with Teacher–Student, and Student–Student Relationships. In Proceedings of the Child & Youth Care Forum. Springer, 2025, pp. 1–30.
64. Lee-Cultura, S.; Sharma, K.; Giannakos, M.N. Multimodal teacher dashboards: Challenges and opportunities of enhancing teacher insights through a case study. *IEEE Transactions on Learning Technologies* **2023**, *17*, 181–201.
65. Yang, W.; Fu, R.; Amin, M.B.; Kang, B. The impact of modern ai in metadata management. *Human-Centric Intelligent Systems* **2025**, pp. 1–28.
66. Mosha, N.F.; Ngulube, P. Metadata standard for continuous preservation, discovery, and reuse of research data in repositories by higher education institutions: A systematic review. *Information* **2023**, *14*, 427.
67. Essa, S.G.; Celik, T.; Human-Hendricks, N.E. Personalized adaptive learning technologies based on machine learning techniques to identify learning styles: A systematic literature review. *IEEE Access* **2023**, *11*, 48392–48409.
68. Lee, Y.; Migut, G.; Specht, M. What attention regulation behaviors tell us about learners in e-reading?: Adaptive data-driven persona development and application based on unsupervised learning. *IEEE Access* **2023**, *11*, 118890–118906.
69. Hussain, T.; Yu, L.; Asim, M.; Ahmed, A.; Wani, M.A. Enhancing e-learning adaptability with automated learning style identification and sentiment analysis: a hybrid deep learning approach for smart education. *Information* **2024**, *15*, 277.
70. Lin, T.C.; Chiu, C.N.; Wang, P.T.; Fang, L.D. VisFactory: Adaptive Multimodal Digital Twin with Integrated Visual-Haptic-Auditory Analytics for Industry 4.0 Engineering Education. In Proceedings of the Multimedia. MDPI, 2025, Vol. 1, p. 3.

 doi:10.20944/preprints202510.2065.v1

71. Salloum, S.A.; Alomari, K.M.; Alfaisal, A.M.; Aljanada, R.A.; Basiouni, A. Emotion recognition for enhanced learning: using AI to detect students' emotions and adjust teaching methods. *Smart Learning Environments* **2025**, *12*, 21.
72. El Maazouzi, Q.; Retbi, A. Multimodal Detection of Emotional and Cognitive States in E-Learning Through Deep Fusion of Visual and Textual Data with NLP. *Computers* **2025**, *14*, 314.
73. Troussas, C.; Krouska, A.; Sgouropoulou, C. Learner Modeling and Analysis. In *Human-Computer Interaction and Augmented Intelligence: The Paradigm of Interactive Machine Learning in Educational Software*; Springer, 2025; pp. 305–345.
74. Sajja, R.; Sermet, Y.; Cikmaz, M.; Cwiertny, D.; Demir, I. Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education. *Information* **2024**, *15*, 596.
75. Gligorea, I.; Cioca, M.; Oancea, R.; Gorski, A.T.; Gorski, H.; Tudorache, P. Adaptive learning using artificial intelligence in e-learning: A literature review. *Education Sciences* **2023**, *13*, 1216.
76. Iliska, D.; Gudoniene, D. Sustainable technology-enhanced learning for learners with dyslexia. *Sustainability* **2025**, *17*, 4513.
77. Szabó, T.; Babály, B.; Pataiová, H.; Kárpáti, A. Development of spatial abilities of preadolescents: What works? *Education Sciences* **2023**, *13*, 312.
78. Gm, D.; Goudar, R.; Kulkarni, A.A.; Rathod, V.N.; Hukkeri, G.S. A digital recommendation system for personalized learning to enhance online education: A review. *IEEE Access* **2024**, *12*, 34019–34041.
79. Rapanta, C.; Botturi, L.; Goodyear, P.; Guàrdia, L.; Koole, M. Online university teaching during and after the Covid-19 crisis: Refocusing teacher presence and learning activity. *Postdigital science and education* **2020**, *2*, 923–945.
80. Nikolic, S.; Daniel, S.; Haque, R.; Belkina, M.; Hassan, G.M.; Grundy, S.; Lyden, S.; Neal, P.; Sandison, C. ChatGPT versus engineering education assessment: a multidisciplinary and multi-institutional benchmarking and analysis of this generative artificial intelligence tool to investigate assessment integrity. *European Journal of Engineering Education* **2023**, *48*, 559–614.
81. Sharif, M.; Uckelmann, D. Multi-Modal LA in Personalized Education Using Deep Reinforcement Learning Based Approach. *IEEE Access* **2024**, *12*, 54049–54065.