Adaptive Learning System Roadmap: Agentic AI and Evidence-Based Learning Gain

Principal AI Architect

October 23, 2025

Contents

1 Executive Summary

The recommended approach is a staged hybrid architecture combining Retrieval-Augmented Generation (RAG) for factual grounding and minimality, and Contextual Bandits (CB) for optimizing content selection based on measured learning gains. This strategy addresses the cold-start challenge while ensuring the system learns to prioritize pedagogical effectiveness over mere relevance.

- **Staged Hybrid Architecture:** Start RAG-first (M1-M3) for rapid prototyping and grounding, then implement Contextual Bandits (M4) for optimization.
- **Minimality Constraint:** Enforce hard limits and prioritize resource segments (1–3 minute video clips, 1–2 page PDF snippets) over whole assets using ASR and semantic segmentation tools (M2).
- **Learning Gain Metric:** Primary metric is $\Delta\theta$, the change in latent learner ability over sequential sessions, measured using Item Response Theory (IRT).
- **Cold-Start Mitigation:** Use heuristics, metadata filters, and ASR/semantic weak labeling to kickstart content recommendations in the absence of historical behavioral data (M1/M2).
- **Agentic Orchestration:** Utilize a Plan-Act-Evaluate loop to dynamically select content, generate formative questions, assess learning, and determine the next optimal action.
- **Agility through PEFT:** Avoid full fine-tuning of large models. Use Parameter Efficient Fine-Tuning (PEFT/LoRA) for smaller, task-specific models (Question Generation, Grading Rubrics) to maintain agility during foundation model updates (M5).
- **Data Leverage:** Pre-train pedagogical components (question difficulty, rubric grading) using existing historical Q&A and assessment data (M3).
- **Measurable Milestones:** Each 2-week milestone delivers an A/B testable increment with clearly defined offline and online metrics (e.g., nDCG, Overkill Rate, Item Discrimination).
- **Scalability:** RAG components are inherently scalable; Bandits provide robust online optimization without requiring massive, expensive RL environments.

2 Architecture Options Comparison

The following table compares at least three viable architecture options against the required constraints and capabilities.

Table 1: Alternative Architecture Options Comparison

Feature	**Option A: RAG + Agentic Orchestration (Baseline)**	**Option C: RL/Bandits on RAG Baseline (Recommended End-State)**	**Option B: LoRA FT for Pedagogy/Style + RAG**
Components	Dense Retrieval, Cross-Encoder Reranker, Agentic Planner (LLM), Prompt Engineering for Pedagogy.	RAG stack + Contextual Bandit Policy (Exploration/Exploitation), Reward Service, Off-Policy Evaluator.	RAG stack + LoRA Adapters for Question Generation/Distractor generation (small LLMs/T5).
Infra	Standard vector DB, serving LLM (e.g., Gemini Pro), high-throughput inference for reranker.	Adds real-time policy server (e.g., Vowpal Wabbit, custom service), extensive telemetry/logging.	Standard RAG + hosting for smaller fine-tuned models/adapters, requiring specialized FT pipelines.
Cost	Moderate (LLM inference is the primary driver).	High Operational Cost (Telemetry, Policy Updates) but potentially high ROI from optimization.	High Setup Cost (Labeling/Training) but potentially lower LLM inference cost if using specialized small LMs.
Latency	Standard RAG latency (0.5s – 2s).	Similar to RAG, plus Policy evaluation (negligible if optimized).	Lower latency for specialized tasks (Q-Gen) if utilizing optimized small models/adapters.
Data Needs	Low (relies on FMs). Needs labeled data for initial cross-encoder training.	High (Requires large volumes of logged user interactions, content choices, and outcome metrics).	High (Requires thousands of high-quality, labeled pedagogical examples).
Cold-Start Viability	Excellent. Depends only on semantic matching and heuristics.	Poor for optimization layer. Must be layered on top of a successful RAG (M4 start).	Moderate. Requires initial exemplar set for FT.

Table 1: Alternative Architecture Options Comparison (Cont.)

Feature	Option A: RAG + Agentic Orchestration (Baseline)	Option C: RL/Bandits on RAG Baseline (Recommended End-State)	Option B: LoRA FT for Pedagogy/Style + RAG
Expected Learning Impact	Good (Factual relevance). Cannot intrinsically optimize for $\Delta\theta$.	Excellent (Directly optimizes content selection for maximal learning gain $\Delta\theta$).	Good (Improves question consistency and quality, leading to better measurement/assessment).
Risks	Prompt drift, reliance on LLM consistency, inability to optimize minimality vs. learning trade-off effectively.	Requires sophisticated causal inference/OPE, risk of exploiting policy to maximize simple metrics (gaming).	High setup cost, risk of adapter incompatibility upon FM upgrade, complexity of managing multiple FT models.

3 Final Recommended Architecture

The recommended approach is the staged implementation of Option C, built on a robust RAG and Agentic foundation (Option A).

3.1 Architectural Components

1. **Learner Interface/Question:** Student asks a question.

2. **Agentic Orchestration (Planner):** Determines the current learner state (θ via IRT model) and plans the next action (Retrieve Resource → Generate Q → Assess).

3. **Retrieval Engine (RAG):**

   - **Chunking/Metadata:** Semantic chunking for PDFs (by section/paragraph) and ASR/Scene Detection for videos (producing 1-3 min segments). Metadata includes duration, Bloom level coverage, prior usage, and difficulty (initial heuristic based on topic).
   - **Hybrid Search:** BM25 (keyword/lexical) + Dense Embedding Search (semantic).
   - **Reranking:** Bi-encoder retrieval (fast initial set) → Cross-Encoder Reranker (BERT-based) for relevance → Maximal Marginal Relevance (MMR) for diversification/sufficiency.

4. **Content Minimization Engine:** Applies hard constraints (e.g., ≤ 3 minutes or ≤ 2 pages). Calculates a "Sufficiency Score" (semantic coverage relative to the question) / Duration to rank for minimality. Selects only the Top-K segments.

5. **Content Selection Policy (Contextual Bandit/RL):** Based on the current learner state (context), selects the single best minimal resource segment to maximize the predicted reward (Learning Gain Proxy).

6. **Pedagogical Layer (Generative Agents):** Uses a smaller LoRA-tuned LLM for consistency.

   - **Question Generation:** Generates N formative questions based on Bloom's Taxonomy, guaranteed to be grounded in the selected resource. Includes distractors and hints.
   - **Rubric-Based Grading:** Assesses student answers against a reference and predefined rubric/ontology, providing structured feedback.

7. **Assessment & Learning Analytics (IRT):** Updates learner latent ability (θ) and question parameters (a, b, c) based on formative assessment results, feeding back into the Agentic Planner.

8. **Telemetry & Feedback Loop:** Logs every interaction, resource choice, and resulting assessment score, providing the data for the Bandit/RL policy updates.

3.2 Architecture Diagram (Textual Representation)

4 Data Plan (Cold-Start to Flywheel)

4.1 Cold-Start for Content Recommendations

Since historical recommendation data is absent, we must rely on heuristics and semantic analysis initially:

1. **Heuristics & Metadata Filters:** Filter content based on essential metadata (Course ID, Topic Tags, Bloom Level appropriateness).

2. **ASR & Semantic Weak Labels:** Use ASR transcription (for video) and section headers (for PDF) combined with semantic embedding coverage to create weak labels indicating which segments of content cover which concepts.

3. **Initial Reranker Training:** Train the initial Cross-Encoder Reranker using synthetic data generated by pairing high-quality existing questions (from historical Q&A logs) with semantically aligned content segments.

4. **Teacher-in-the-Loop (M1/M2):** In the prototype phase, implement an initial low-volume feedback loop where subject matter experts (SMEs) score the minimality and relevance of the top-5 resource suggestions for a random sample of queries.

```
[Learner Question] -> [Agentic Planner]
                               | ( state, plan)
                               V
+-------------------------------------------------------+
| 1. Retrieval Engine (Hybrid Search + Cross-Reranker)  |
+-------------------------------------------------------+
                               | (Top-N segments)
                               V
+-------------------------------------------------------+
| 2. Content Minimization (Sufficiency/Duration Score)  |
+-------------------------------------------------------+
                               | (Minimal Candidates)
                               V
+-------------------------------------------------------+
| 3. Content Selection Policy (Contextual Bandit)       |
|    Reward:  Proxy + Minimality Bonus                |
+-------------------------------------------------------+
                               | (Selected minimal resource)
                               V
[Learner views resource] -> [4. Pedagogical Layer (Q-Gen LoRA)]
                               |
                               V
[Formative Assessment] -> [5. Assessment Layer (IRT Model)]
                               | (New , Item Parameters)
                               V
[Telemetry Log/Reward Service] -> [Bandit Policy Update]
```

Figure 1: Recommended Adaptive Learning Architecture Flow

4.2 Rapid Dataset Creation and Labeling

1. **Offline Labeling Protocols:** Define strict rubrics for ”Best Minimal Resource” (e.g., must cover X concepts, must not exceed Y duration, scored 1-5). Use SMEs for high-precision labeling.

2. **Active Learning Loops:** Once the RAG is live, use user behavior (CTR, minimal resource consumption time, subsequent quiz performance) as proxy labels. Prioritize labeling for queries where the RAG/Bandit policy uncertainty is highest (active learning).

3. **Inter-Rater Agreement (IRA):** Ensure high IRA ($\kappa > 0.7$) among SMEs used for labeling core Reranker and Pedagogical samples.

4.3 Leveraging Existing Q&A/Assessment Logs

1. **Question Generation Refinement:** Use historical Q&A pairs (high-quality questions + canonical answers) as exemplars for few-shot prompting and later LoRA fine-tuning (M3/M5). This ensures generated questions align with the established pedagogical style.

2. **IRT Parameter Calibration:** Use historical assessment data (item responses) to pre-calibrate initial Item Response Theory (IRT) parameters (a, b, c). This provides a stable baseline for measuring learner ability (θ) from M3 onwards.

3. **Grading Rubrics:** Extract common errors and correct answer variations from historical student responses to refine the automatic rubric grader for better precision.

5 Metrics & Evaluation

Metrics must clearly distinguish between technical performance, efficiency, and true learning outcomes.

5.1 Technical & Efficiency Metrics (M1, M2)

- **Retrieval/Selection:** nDCG@k (Normalized Discounted Cumulative Gain), Mean Average Precision (MAP), Recall@k.
- **Coverage:** Percentage of user questions for which the RAG system finds at least one highly relevant resource (Relevance Score > 0.8).
- **Latency:** Time-to-first-useful-resource (P95 latency of the full RAG pipeline).
- **Minimality:** Median resource length (Target: ≤ 3 min or ≤ 2 pages). "Overkill Rate" (% of suggestions exceeding target length). Compression Ratio (Length of suggested segment / Length of full asset).

5.2 Pedagogical & Assessment Quality Metrics (M3)

- **Question Quality (Offline):** Expert rubric scores (Clarity, Alignment to Resource, Bloom Level). Pass@k on canonical answers (automatic verification against ground truth). Factuality via Reference-Grounded checks.
- **Assessment Quality (IRT):**
  - Item Discrimination (a): Measures how well an item distinguishes between high and low-ability students. Target $a > 0.8$.
  - Item Difficulty (b): Ensures generated questions cover an appropriate difficulty range.
  - Ability (θ) Metrics: Stability (low standard error of measurement, SE) and Predictive Validity (correlation between θ and downstream course grade).

5.3 Learning Outcome Metrics (M4, M5, M6)

These are the primary success metrics for the entire system.

- **Latent Ability Change $(\Delta\theta)$:** The primary metric. Measured as the mean or median difference in estimated learner ability (θ) between the start of a session and the end, after consuming the resource and completing formative questions.
- **Mastery Progression:** Time/sessions required for learners to move from Novice $(\theta < -1)$ to Proficient $(\theta > 0)$ in specific topics.
- **Normalized Gain (g):** $g = (\text{Posttest Score} - \text{Pretest Score})/(100 - \text{Pretest Score})$. Use sequential quizzes/assessments as Pre/Post proxies.

- **Downstream Performance:** Correlation between system usage/mastery and final course grades or standardized test scores.

5.4 Engagement and Safety Metrics

- **Engagement (Secondary):** Resource Consumption Rate (RCR), Click-Through Rate (CTR) for suggestions, Dwell Time. (Used as feature inputs, not success criteria).
- **Safety/Accuracy:** Hallucination Rate (vs. reference content), Contradiction Rate, Refusal/Deferral Accuracy (when the agent correctly refuses to answer or suggests external content).

6 Stepwise Roadmap (12 Weeks)

The plan is designed for shippable increments, allowing for A/B testing and decision points at the end of each phase.

Table 2: 12-Week Adaptive Learning Roadmap

Milestone	Key Tasks & Focus	Acceptance Criteria & Metrics
M1 (Wks 1–2): RAG Baseline & Minimality Hard Caps	1. Deploy basic RAG (Dense embedding + BM25) for retrieval. 2. Implement Content Minimization: Hard-cap all resource suggestions to 5 minutes/3 pages maximum. 3. Define chunking strategy and initial metadata schema. 4. Implement strict JSON-structured outputs for agent communication. 5. Offline Evaluation: Red team cases for irrelevant suggestions.	1. nDCG@5 ≥ 0.65 (offline evaluation using synthetic/SME labels). 2. Overkill Rate $\leq 5\%$ (hard cap violations). 3. P95 Retrieval Latency $\leq 1.5s$. 4. Decision Point: Proceed to advanced RAG or optimize embedding choice.
M2 (Wks 3–4): Advanced Retrieval & Segmentation	1. Implement Cross-Encoder Reranking on Top-50 candidates. 2. Integrate Video/PDF Segmenter (ASR for clips, section detection for PDF). 3. Introduce Sufficiency Score (semantic coverage / duration) ranking. 4. Set up telemetry logging for resource CTR and minimality metrics.	1. nDCG@1 improves by 10% vs. M1 baseline. 2. Median Resource Length $\leq$ 3 minutes. 3. First live A/B test (M2 RAG vs. M1 baseline) tracking CTR/RCR. 4. Decision Point: Verify data quality for resource segments.

Table 2: 12-Week Adaptive Learning Roadmap (Cont.)

Milestone	**Key Tasks & Focus**	**Acceptance Criteria & Metrics**
M3 (Wks 5–6): Pedagogy Tools V1 & Assessment Baseline	1. Implement Prompted Question Generator (Q-Gen) grounded strictly in the suggested resource. 2. Develop Rubric Grader and Hinting features using few-shot exemplars from historical Q&A. 3. Deploy initial IRT model (pre-calibrated with historical data) to track θ. 4. Agentic Orchestration V1: Full Plan → Retrieve → Q-Gen → Assess loop operational.	1. Question Quality Score (SME Rubric) $\geq$ 4.0/5.0. 2. IRT Model calibration stability checked (SE for θ is low). 3. Live A/B test: M3 (full Q-Gen/Assessment) vs. M2 (no Q-Gen) tracking practice opportunities per session.
M4 (Wks 7–8): Bandit Optimization for Learning Gain	1. Deploy Contextual Bandit (Thompson Sampling/UCB) policy server on top of M3 RAG. 2. Define and deploy Reward Service: $R = w_1$(Quiz Correctness Uplift) + w_2(Minimality). 3. Begin logging policy diagnostics and data for Off-Policy Evaluation (OPE). 4. Implement Safety/Guardrails: Source citation checks, JSON validation.	1. Offline OPE demonstrates $\geq$ 5% uplift potential in expected reward compared to RAG baseline. 2. Bandit Exploration Rate is calibrated (e.g., 10%). 3. Live A/B test: M4 Bandit Policy vs. M3 RAG Baseline, tracking $\Delta\theta$ per session.
M5 (Wks 9–10): LoRA Fine-Tuning & Refinement	1. Use high-quality labeled data (M1-M4 collection) to train LoRA adapters for Q-Gen and Distractor Quality (Consistency/Pedagogical Tone). 2. Implement adapter switching strategy for model upgrades. 3. Optimize Cross-Encoder for faster inference or replace with more compact architecture.	1. Q-Gen Consistency (Style/Tone) improves by 20% (measured by human/LLM evaluator). 2. Inference latency for Q-Gen tasks is reduced. 3. Live A/B test: Prompt-only Q-Gen vs. LoRA-tuned Q-Gen, tracking item quality and cost/latency.

Table 2: 12-Week Adaptive Learning Roadmap (Cont.)

Milestone	**Key Tasks & Focus**	**Acceptance Criteria & Metrics**
M6 (Wks 11–12): Production Hardening & Scaling	1. Implement continuous monitoring, bias checks, and compliance filters (PII removal, accessibility). 2. Production-grade telemetry dashboards for all operational and learning metrics (θ drift, bandit diagnostics). 3. Finalize Educator Dashboard features for policy oversight and content performance.	1. System uptime $\geq$ 99.9% for core inference services. 2. Zero PII/Compliance violations found in final audit. 3. Final decision memo prepared on go/no-go for scaling based on M4/M5 $\Delta\theta$ results.

7 Reinforcement Learning Design (Practical)

The goal is to optimize the policy (the content selection engine) to maximize measurable learning outcomes while maintaining efficiency constraints.

7.1 Reward Function Definition

Since true learning gain ($\Delta\theta$) is a latent metric updated post-assessment, we define a composite, multi-objective reward (R_t) for selecting resource r at time t:

$$R_t = w_1 \cdot \underbrace{\text{Quiz Correctness Uplift}}_{\text{(proxy for } \Delta\theta\text{)}} + w_2 \cdot \underbrace{\text{Minimality Bonus}}_{\text{(duration/pages)}} - w_3 \cdot \underbrace{\text{Irrelevance Penalty}}_{\text{(low CTR/Skip)}} - w_4 \cdot \underbrace{\text{Cost/Latency Pe}}_{\text{(efficiency)}}$$

- **Quiz Correctness Uplift (Proxy):** Score on the formative quiz immediately following the resource consumption. This serves as a rapid, high-frequency signal correlated with $\Delta\theta$. Weight w_1 is highest.
- **Minimality Bonus:** A scalar bonus inversely proportional to the segment duration, up to the hard cap.
- **Irrelevance Penalty:** Applied if the student skips the resource or fails the subsequent quiz significantly (suggesting irrelevance).

7.2 Implementation Strategy: Contextual Bandits First

1. **Start with Contextual Bandits (M4):** Implement Thompson Sampling or UCB. Bandits are less data-hungry than full RL and excel at exploring the content selection space (Arms = Top-K minimal resources) based on immediate context (Learner θ, question type, recent performance).

2. **Context Features:** The context vector includes Learner θ, Item parameters (difficulty of the current question topic), and features of the candidate resources (segment duration, source popularity, predicted relevance score from the Reranker).

3. **Off-Policy Evaluation (OPE):** Essential for responsible iteration. Use Inverse Propensity Scoring (IPS) or Doubly Robust (DR) estimators to evaluate new policies (e.g., M5 Bandit Policy) offline using historical logged data before deployment, minimizing risk.

4. **Escalation to Full RL:** Only transition to a full Reinforcement Learning setup (e.g., DQN or A2C) if the Contextual Bandit performance plateaus, and if we have successfully built a high-fidelity learning simulator (based on IRT parameter tracking and user behavior models) that allows for safe, large-scale training.

8 Fine-tuning Policy (When and How)

The primary policy is to prioritize prompt engineering and RAG for core factual tasks, reserving fine-tuning for tasks requiring high consistency and style, ensuring agility.

8.1 Task-Specific LoRA/PEFT Focus (M5)

We will use LoRA (Low-Rank Adaptation) or similar PEFT methods on smaller, task-specific models (e.g., T5 or a smaller Gemini model) rather than full fine-tuning the primary large language model.

1. **Pedagogical Consistency (Question Generation/Hinting):** Fine-tuning improves the consistency of question style, Bloom level alignment, and hint quality, reducing reliance on complex, long prompts.

2. **Distractor Generation Quality:** Fine-tuning ensures distractors are plausible but incorrect, based on common learner misconceptions derived from historical data.

3. **Rubric-Based Grading:** Fine-tuning to improve the accuracy of structured output grading against a complex internal rubric ontology.

8.2 Fine-Tuning Decision Checklist (Entry Criteria)

Fine-tuning is approved only if ALL of these criteria are met for a specific task:

- **Performance Plateau:** Demonstrated inability of advanced prompt engineering (including Chain-of-Thought) to meet quality/consistency targets.
- **Data Availability:** $\geq 5,000$ high-quality, labeled exemplars per task (e.g., Question $\rightarrow$ LoRA $\rightarrow$ High-Quality Output).
- **Inference Savings/Latency:** Projected inference cost reduction or latency improvement justifies the training and serving overhead.
- **High Consistency Need:** The task requires absolute stylistic consistency (e.g., pedagogy tone, specific output format).

8.3 Model Upgrade Migration Plan

Using LoRA adapters ensures agility. If Google releases a new base model (e.g., Gemini 3), the migration strategy is:

1. Test the performance of the new base model with existing prompt-only systems (M1-M4).

2. Attempt adapter transfer: If the new base model is architecturally compatible, transfer the existing LoRA weights.

3. Retrain only the small LoRA adapter weights (not the full model) on the new base model if transfer fails or performance degrades. This is significantly faster and cheaper than full fine-tuning.

9 Risks & Mitigations

Table 3: Key Risks and Mitigation Strategies

Risk	**Mitigation**
Cold-Start for content recommendations.	Implement heuristics and weak labeling (ASR + semantic coverage) immediately (M1/M2). Use SME review (Teacher-in-the-Loop) to generate initial high-quality labels for the Cross-Encoder.
Policy (Bandit) exploits simple metrics.	Use a multi-objective reward function with w_1 (Learning Gain proxy) as the dominant weight. Use IRT $\Delta\theta$ as the final, long-term truth metric, which is harder to game than simple correctness.
Over-long resources/Overkill.	Enforce strict hard caps (3-minute clips). Use the Sufficiency Score / Duration ratio as a hard ranking factor (M2). MMR ensures diverse coverage without excessive length.
Hallucinations (in Q-Gen or feedback).	Mandate Retrieval-Grounded Verification (RGV): Every generated question/hint must be traceable and verified against the suggested minimal resource content. Implement an answerability checker using the resource segment.
Misaligned Difficulty (IRT drift).	Establish a regular IRT recalibration cadence (e.g., monthly). Use anchor items (standard questions) to detect and correct parameter drift. Monitor item discrimination (a) as a quality filter.
Data Logging Volume/Privacy.	Implement strict PII filtering and anonymization at the logging layer (M6). Ensure role-based access to telemetry. Explore deployment options that maintain data within institutional boundaries.

10 Deliverables per Milestone

The following is a list of concrete, verifiable artifacts delivered at the completion of each milestone.

- **M1 (RAG Baseline):** Retrieval Pipeline Notebooks, Initial Embeddings and Vector DB Configuration, Hard-Cap Logic Code, Offline nDCG Evaluation Report, Red Team Stress Test Cases.
- **M2 (Advanced Reranking/Segmentation):** ASR/PDF Segmenter Pipeline, Cross-Encoder Training Data Card and Model Card, Sufficiency Score Implementation, Telemetry Dashboards V1 (CTR, Length).
- **M3 (Pedagogy/Assessment):** Q-Gen Prompt Library (version controlled), Rubric Grader Logic, IRT Model Artifacts (pre-calibrated parameters), Full Agentic Orchestration Flowchart, Live A/B Test Design Document.
- **M4 (Bandit Optimization):** Contextual Bandit Policy Server Code, Reward Service Implementation (including w_1, w_2 values), Off-Policy Evaluation (OPE) Recipe and Initial Report, Live $\Delta\theta$ Tracking Dashboard.
- **M5 (LoRA Fine-Tuning):** Labeled Dataset Card for Q-Gen FT, LoRA Adapter Weights and Model Card, Adapter Transfer/Migration Plan, Inference Latency Benchmark Report (FT vs. Prompt-Only).
- **M6 (Production Hardening):** Safety Guardrail Implementation (Contradiction checks, refusal paths), Compliance Audit Report (PII/Bias), Final Production Telemetry Dashboards, Decision Memo for Scaling.