Evaluating Adaptive Personalization of Educational Readings with Simulated Learners

Ryan T. Woo* **Anmol Rao*** **Aryan Keluskar** **Yinong Chen**

School of Computing and Augmented Intelligence
Arizona State University
{rtwoo,arao75,akeluska,yinong.chen}@asu.edu

Abstract

We present a framework for evaluating adaptive personalization of educational reading materials with theory-grounded simulated learners. The system builds a learning-objective and knowledge-component ontology from Wikibooks open textbooks, curates it in a browser-based Ontology Atlas, labels textbook chunks with ontology entities, and generates aligned reading–assessment pairs. Simulated readers learn from passages through a Construction–Integration-inspired memory model with DIME-style reader factors, KREC-style misconception revision, and an open New Dale–Chall readability signal. Answers are produced by score-based option selection over the learner's explicit memory state, while BKT drives adaptation. Under this explicit simulator parameterization and three sampled subject ontologies, adaptive reading significantly improved outcomes in computer science, yielded smaller positive but inconclusive gains in inorganic chemistry, and was neutral to slightly negative in general biology.

1 Introduction

With more students entering higher education, instructors increasingly face a familiar mismatch: students are expected to learn the same material at the same pace even though their understanding of particular concepts can vary widely. In an introductory course, one student may already understand a subset of the target concepts, while another may still hold robust misconceptions about those same concepts. When both students are given identical readings, examples, and explanations, one may be under-challenged while the other is overwhelmed. Addressing this in a real classroom typically requires instructors to rewrite materials, manually identify which concepts students struggle with, and provide individualized support. Those interventions are difficult to scale.

Intelligent tutoring systems (ITSs) aim to address this problem by adapting instruction to the learner. Recent reviews suggest that ITSs often improve learning and performance, but that their impact depends strongly on the instructional setting, the adaptation mechanism, and the way systems are evaluated (Huang et al., 2025; Létourneau et al., 2025). In many ITS settings, however, the core intervention is an item sequence, a hint policy, or post-hoc feedback. Our setting is different: the primary intervention is *reading material*. Questions are still important, but mainly because they reveal evidence about what the learner knows after reading and provide observations for mastery tracking.

To evaluate whether adaptive reading policies help, we need simulated learners that can (i) learn from text, (ii) differ in reading-related characteristics, (iii) preserve misconceptions rather than merely lacking knowledge, and (iv) later answer multiple-choice questions from what they remember rather than from unrestricted access to the source passage. Simulated learners have long been proposed as tools for teachers, for learners, and for instructional designers conducting formative evaluation (VanLehn et al., 1994). More recently, a systematic review across AIED-related fields found uses spanning teacher training, adaptive algorithm development, virtual collaborators, and testing of learning environments, while also concluding that realism and validity are often under-tested (Käser and Alexandron, 2024). Off-the-shelf LLM prompting is a poor fit for that requirement because powerful LLMs often answer beyond the intended ability of a partially knowledgeable learner. Yuan et al. (2026) characterize this problem as a *competence paradox* and argue for constrained student simula-

*equal contribution.

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 941–950

tion under an explicit epistemic state specification.

We therefore present a theory-grounded simulated reader for adaptive reading personalization. The system builds a learning-objective (LO) and knowledge-component (KC) ontology from open textbooks, tracks KC mastery with BKT (Corbett and Anderson, 1994), generates ontology-driven reading and assessment variants, and simulates reading via a comprehension model grounded in the Construction–Integration (CI) theory of discourse comprehension (Kintsch, 1988). Learner heterogeneity is parameterized with DIME-style reader factors (Cromley and Azevedo, 2007; Cromley et al., 2010), misconceptions are revised through the Knowledge Revision Components framework (KREC) (Kendeou and O'Brien, 2014; Kendeou, 2024), and later answer selection is performed through score-based option selection over an explicit epistemic boundary.

The experiments focus on offline simulation so that the instructional policy can be studied before classroom deployment. This setup compares adaptive and non-adaptive reading policies under matched simulated cohorts while keeping the content pipeline, question format, and mastery tracker fixed.

Our main contributions are as follows:

- We formulate adaptive reading personalization as a closed-loop problem in which reading is the primary intervention and assessment is the observation channel for KC-level mastery tracking.
- We present an ontology-grounded content pipeline that links open textbook chunks, reading variants, and assessment items to shared LOs and KCs.
- We develop a simulated reader that combines CI, DIME, KREC, and an open New Dale–Chall readability signal to model what a learner comprehends, revises, retains, and later retrieves.
- We report multi-subject experimental results showing that adaptive reading yields domain-dependent gains: statistically reliable improvements in the sampled computer-science setting, positive but inconclusive gains in inorganic chemistry, and neutral to slightly negative outcomes in general biology.

2 Related Work

2.1 Adaptive tutoring and knowledge tracing

Knowledge tracing estimates latent mastery from observed responses, and BKT remains attractive because it is interpretable at the level of individual KCs (Corbett and Anderson, 1994). Although deep models can improve prediction, interpretability remains important for deployment and diagnosis (Ding and Larson, 2021). We therefore use BKT so that the state driving adaptation remains understandable.

Most ITS personalization focuses on problem selection, hints, or feedback. By contrast, we center on explanatory reading material. Because reading can change knowledge without explicit responses, we separate the learner's hidden *true* knowledge from the tutor's BKT-based *estimate*, and only the latter drives adaptation.

2.2 Simulated learners in educational technology

Simulated learners have long been used in educational technology. VanLehn et al. (1994) categorize their use for teachers, learners, and instructional designers; our setting aligns most closely with formative evaluation of instructional policies before deployment.

Recent work highlights broader uses, including teacher training, hypothesis generation, adaptive algorithm development, virtual collaborators, and environment testing, while also noting that many simulators remain under-validated for their intended decisions (Käser and Alexandron, 2024). This motivates our emphasis on explicit learner state, inspectable memory, and theory-grounded reading dynamics.

2.3 Reading comprehension and computational reader models

Our model is grounded in the Construction–Integration framework (Kintsch, 1988), where readers form a propositional *textbase* and a richer *situation model* integrating prior knowledge. This supports modeling partial retention, revision, and distortion of information.

We extend CI with DIME, which captures reader differences in background knowledge, vocabulary, and inference (Cromley and Azevedo, 2007; Cromley et al., 2010), and KREC, which models revision of incorrect prior knowledge (Kendeou and O'Brien, 2014; Kendeou, 2024).

This matters because many systematic wrong answers reflect stable misconceptions rather than random noise (van den Broek and Kendeou, 2008; Gierl et al., 2017).

Prior work has implemented CI-like automated readers (Lemaire et al., 2006); our contribution is to embed such a reader in a KC-level adaptive instructional loop.

2.4 Epistemic boundaries and answer modeling

Recent work argues that valid student simulation should expose an explicit epistemic state and bounded competence (Yuan et al., 2026). We adopt that principle, but do not use free-form generation for answer choice. Instead, answers are selected by scoring options from the learner's memory state, including mastery, misconceptions, retrieved traces, and item difficulty. This design is efficient, auditable, and consistent with constrained learner behavior. It also favors inspectable state and reproducible repeated simulations over unconstrained generative answers.

2.5 Readability and reader–text matching

Reader–text fit is important but should remain simple to operationalize. We use the New Dale–Chall readability formula (Chall and Dale, 1995), computed via `textstat` (Bansal and Aggarwal, 2026), as a reproducible measure of text difficulty combined with learner parameters. The formula is treated only as a lightweight signal; its tradeoffs are discussed in the limitations.

3 System Overview

The overall system converts open educational content into a grounded personalization loop. The pipeline can be summarized in five stages: (1) ontology construction and authoring, (2) chunk labeling and retrieval indexing, (3) ontology-driven reading and assessment generation, (4) theory-grounded simulated reading and score-based answer selection, and (5) BKT-based adaptation. Figure 1 separates the fixed source-corpus, ontology, grounding, and generation stages from the online simulation. Figure 2 shows the simulated reading and adaptation loop as a closed cycle.

The experiments use Wikibooks source textbooks: *Foundations of Computer Science*, *General Biology*, and *Introduction to Inorganic Chemistry* (Wikibooks contributors, 2026a,b,c). These books serve as course-specific corpora: the ontology, retrieval index, generated readings, and assessments are grounded in the same instructional source material.

3.1 Ontology Atlas and ontology construction

We first build an ontology of learning objectives and knowledge components from open textbooks. Each LO maps to one or more KCs. The ontology is created with an LLM-assisted workflow that proposes candidate LOs and KCs from source material and then consolidates them into a course-specific YAML schema. We use the ontology as a common representational layer for retrieval, generation, mastery tracking, and simulation.

The ontology construction process is semi-automated rather than fully automatic. LLM-generated LO and KC candidates are treated as proposals that can be accepted, merged, split, renamed, or deleted in Atlas before experiment artifacts are generated. Merge decisions are appropriate when two KCs represent the same assessable concept at the same grain size; split decisions are appropriate when a proposed KC combines multiple independently assessable skills or concepts. In the reported workflow, ontology files are fixed within each subject-level adaptive-control comparison so that the two policies use the same ontology.

To make this layer inspectable, we built Ontology Atlas, a browser-based ontology workspace backed directly by the YAML files used elsewhere in the system. Atlas exposes a coverage inventory showing chapter, LO, and KC counts; a searchable navigator over IDs, statements, labels, and descriptions; a D3 relationship graph for the chapter $\rightarrow$ LO $\rightarrow$ KC hierarchy; and a details panel for editing entries, importing/exporting YAML, validating fields, and saving back to disk. Atlas is therefore not just a visualization layer; it is the authoring interface for the ontology that the generation and experiment services consume.

In the current codebase, Ontology Atlas surfaces multiple full-course ontologies, including the three used in the experiments reported here: foundations of computer science (16 chapters, 53 LOs, 131 KCs), general biology (20 chapters, 60 LOs, 172 KCs), and introductory inorganic chemistry (12 chapters, 57 LOs, 177 KCs). This coverage view was useful when selecting sampled LOs for the experiments and when checking whether KC granularity was comparable across subjects.

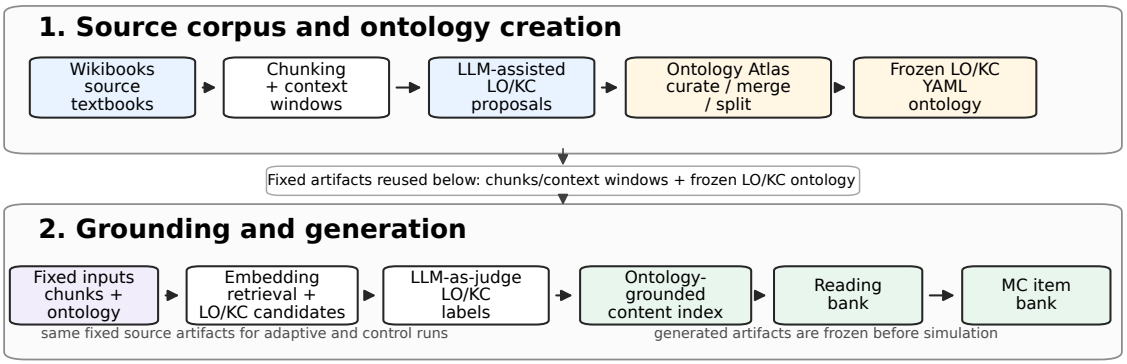

Figure 1: Fixed source-corpus, ontology, grounding, and generation pipeline used by both conditions.

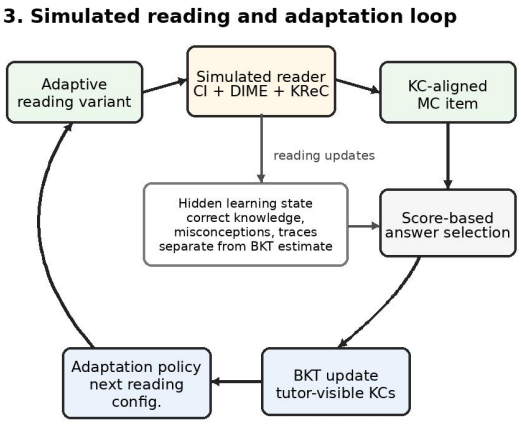

Figure 2: Closed simulated reading and adaptation loop.

3.2 Chunk labeling and retrieval indexing

The textbook corpus is segmented into fine-grained chunks and coarser context chunks. Fine-grained chunks serve as the primary evidence units for labeling, while coarser chunks provide broader context during downstream generation. Each fine-grained chunk is labeled with one or more LOs and KCs using a three-stage LLM-as-judge pipeline:

1. encode the chunk as a semantic embedding,
2. retrieve candidate LO and KC labels with nearest-neighbor search in a vector database, and
3. prompt an LLM to adjudicate among the candidates and return the final label set.

The result is a grounded index in which every retained content unit is tied to the ontology and remains linked to its surrounding context. Because Atlas edits the same YAML that the labeling and generation services read, ontology revisions immediately propagate to retrieval and generation.

3.3 Ontology-driven reading and assessment generation

Using the labeled corpus, we generate paired readings and assessments that comprehensively cover one LO at a time. Each generated reading is grounded in retrieved fine-grained chunks and optional coarser context chunks; generation parameters control explanation depth, example density, misconception refutation, and target difficulty. Each assessment is a fresh isomorphic multiple-choice variant aligned to the same LO and KCs rather than an exact reuse of the same item surface form.

This choice is important for two reasons. First, exact question reuse would make policy evaluation more sensitive to item wording than to learner state. Second, our simulated learners do not learn from assessment interactions; only reading updates their hidden semantic state. Using fresh isomorphic variants therefore preserves comparability across iterations without allowing assessment exposure to become an unintended instructional channel.

For preprocessing, semantic embeddings used `qwen3-embedding-8b`; LLM-assisted ontology proposal, chunk-label adjudication, reading generation, and assessment generation used `qwen3-30b-a3b-instruct-2507`. Once generated, the ontology, labels, readings, and assessments are treated as fixed artifacts for the adaptive-control comparison.

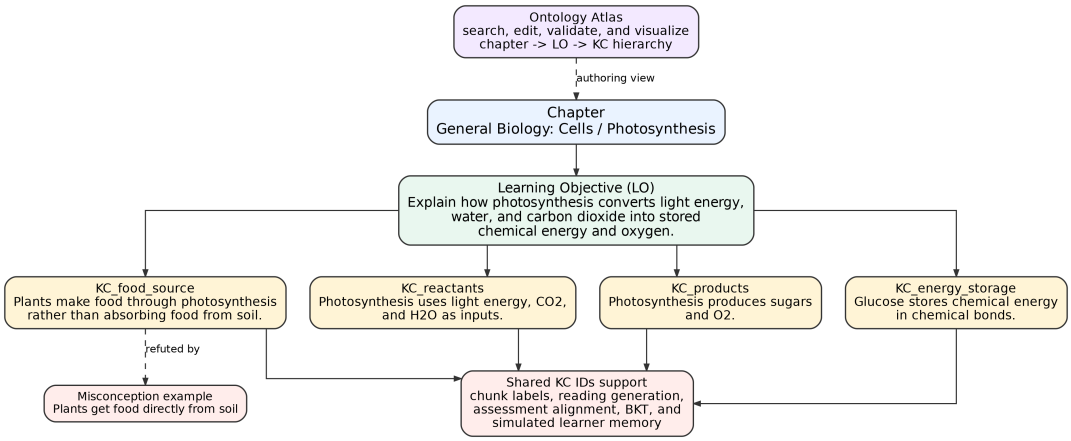

Figure 3: Example chapter → LO → KC relation visualized by Ontology Atlas for photosynthesis.

4 Methodology

4.1 Separating hidden learning from observed mastery

The simulated learner maintains two distinct states. The first is a hidden *true learning state* that governs what the learner actually knows, misunderstands, and remembers. The second is the tutor-visible BKT estimate that is updated only from observed answers. This separation prevents the simulation from becoming self-fulfilling: the personalization policy acts on what the tutor *infers*, not on hidden ground truth.

For learner i and KC k, we represent hidden state as

$$z_i = \{m^*_{i,k},\ w_{i,k,1:M},\ r_i,\ s_i,\ e_i\}, \qquad (1)$$

where $m^*_{i,k}$ is the strength of correct knowledge for KC k, $w_{i,k,1:M}$ are strengths of KC-specific misconceptions, r_i is a reader profile, s_i is a revision-and-response profile, and e_i stores recent episodic traces from readings. The tutor only observes a separate KC mastery estimate $\hat{m}_{i,k}$ maintained by BKT.

4.2 Reader profiles and readability matching

We do not use vague "learning styles." Instead, learner heterogeneity is parameterized through theory-grounded reader profiles. Following DIME, each learner has parameters for background knowledge, vocabulary, inferencing, and strategy use (Cromley and Azevedo, 2007; Cromley et al., 2010). Because the cohort is intended to approximate college freshmen who have completed secondary school, basic decoding is assumed to be near ceiling; the meaningful variability is in comprehension-relevant factors and in misconception prevalence.

We combine these learner-side parameters with a reader–text fit term based on the New Dale–Chall score. The score is used as a compact, reproducible difficulty signal alongside the richer learner-side profile.

For passage j, let d_j be the readability score computed with `textstat`. Let a_i be a scalar readability-ability parameter for learner i. We define

$$\text{match}_{ij} = 1 - \frac{|a_i - d_j|}{6}, \qquad (2)$$

clipped to $[-1, 1]$, so higher values indicate a better match between learner and text. This term does not replace the richer reader profile; it provides a compact difficulty signal that can be integrated into the comprehension update.

4.3 From passages to proposition-like teaching events

We model reading at the level of comprehension and memory, not time. Each generated passage is segmented into short teaching events, which serve as proposition-like units for the simulation. The current implementation uses a lean operational schema rather than full semantic parsing: each event stores its surface text, linked KC IDs, a clarity score, a refutation-strength score, and a refresh flag.

These fields have direct roles in the simulation. *KC links* determine which latent mastery variables and misconception weights the event can update. *Clarity* approximates how easy the event is to encode; in the current implementation it is estimated from event length, so shorter events are treated

State component	Meaning	Role in the simulation
$m^*_{i,k}$	Hidden strength of correct knowledge for KC k	Governs durable support for correct answers and future learning from related readings
$w_{i,k,m}$	Hidden strength of misconception m for KC k	Attracts misconception-consistent distractors and resists revision when refutation is weak
r_i	Reader profile (background knowledge, vocabulary, inferencing, strategy use, readability ability)	Determines how well teaching events are encoded from text
s_i	Revision and response profile (skepticism, revision willingness, detail preference, noise)	Controls how new text changes prior beliefs and how answers are emitted
e_i	Episodic traces of recently read teaching events	Supports short-term retrieval even when durable KC mastery is still weak
$\hat{m}_{i,k}$	Tutor-visible BKT mastery estimate	Drives reading adaptation but does not directly change hidden learning

Table 1: Learner-state variables.

as easier to encode than longer ones. *Refutation strength* approximates how explicitly the event confronts a likely misconception; in the current implementation it is triggered by lexical cues such as `misconception`, `incorrectly`, `rather than`, `instead of`, or `do not`, and it scales misconception reduction during learning. *Refresh* marks lower-gain reinforcement events when review KCs are included in a passage; operationally, refresh is a lightweight heuristic rather than a discourse parser.

This representation keeps learner memory close to the CI notion of a textbase while remaining simple enough to generate, score, and inspect at scale. Encoded events create episodic traces that preserve the event text, KC links, clarity, refutation strength, readability score, and later distortion, making subsequent answer behavior inspectable. When provenance is needed, it is retained at the reading-asset level through source chunk identifiers rather than per-event source spans.

For example, the sentence pair "Plants do not get food from soil. They make food through photosynthesis." is stored as a proposition-like teaching event linked to a photosynthesis KC:

```
{
proposition_id: "prop_001",
text: "Plants do not get food from soil.
They make food through photosynthesis.",
kc_ids: {"KC_photosynthesis_food_source"},
clarity: 0.92,
refutation_strength: 1.00,
is_refresh: false
}
```

The record is operational rather than a full semantic proposition.

4.4 Comprehension as textbase construction and situation-model integration

The reading update proceeds in two stages. First, the learner encodes teaching events into a textbase-like episodic memory. Second, the learner integrates those events with prior knowledge to update correct knowledge and misconception weights. Operationally, the encoding strength of event p for learner i reading passage j is modeled as

$$\begin{aligned}\text{enc}_{i,p} = \sigma\big(&\alpha + 1.1\,\text{clarity}_p + 0.55\,\text{match}_{ij} \\ &+ \beta^\top r_i + 0.3(1-\bar{m}_{i,p}) \\ &+ 0.35\,\text{attention}_i\big).\end{aligned} \tag{3}$$

Here, $\bar{m}_{i,p}$ is mean prior mastery over the event's target KCs and $\beta^\top r_i$ collects vocabulary, inference, strategy, and background-knowledge effects. If an event is encoded, it creates an episodic trace whose later activation depends on recency, rehearsal, lexical overlap with the question, and distortion noise.

Once encoded, an event updates KC state by increasing correct knowledge and reducing misconception weight. In the implementation, correct-knowledge gain is stronger for clearer events, better reader–text matches, and learners with higher revision willingness and detail preference. Misconception reduction is scaled by the event's refutation strength, consistent with KREC's emphasis on how prior incorrect knowledge is confronted during reading (Kendeou and O'Brien, 2014; Kendeou, 2024). Refresh events receive a smaller gain multiplier than new target events.

The coefficients in Equation 3 and Equation 4 were selected during simulator development and

then held fixed across subject comparisons. They encode qualitative assumptions rather than fitted human-data parameters: clarity, match, relevant prior knowledge, attention, and strategy use increase encoding; misconception strength and item difficulty reduce correct responding; and retrieved traces provide partial support.

4.5 Score-based answer option selection from learner memory

At assessment time, the learner does *not* receive the original passage. Instead, the simulator retrieves relevant traces from memory and computes a score for each item from hidden mastery, misconception weight, retrieved-trace support, and response-style variables. The guiding principle is still epistemic boundedness in the sense of Yuan et al. (2026): the learner answers only from what it has encoded and retained. However, the current implementation uses score-based option selection rather than free-form generative reasoning.

For item q, let $\bar{m}_{iq}$ be mean correct knowledge over the target KCs, $\bar{w}_{iq}$ the mean misconception weight, and τ_{iq} the mean activation of retrieved traces. The item's response utility is

$$\begin{aligned} u_{iq} = & -b_q + 2.2\,\bar{m}_{iq} + 0.38\,\tau_{iq} \\ & - 1.25\,\bar{w}_{iq} + 0.35\,\text{attention}_i \\ & - 0.25\,\text{guess}_i, \end{aligned} \tag{4}$$

where b_q is a difficulty offset derived from the item's difficulty band. The probability of a correct answer is then

$$P(y_{iq} = 1) = \sigma(u_{iq}), \tag{5}$$

clipped to a small interior interval to avoid degenerate probabilities.

If the learner answers incorrectly, the selected distractor is *not* sampled uniformly. Instead, distractors are scored by lexical overlap between the retrieved trace text and the distractor text/rationale, plus learner-specific guessing bias. This makes misconception-consistent wrong options more likely than random errors when the learner's memory traces support them. The simulator also records an explicit epistemic-state summary for each response, including target KCs, correct-knowledge support, misconception support, and the IDs of the retrieved traces used during scoring.

Subject	Ch.	LOs	KCs
Computer Science	16	53	131
General Biology	20	60	172
Inorganic Chemistry	12	57	177

Table 2: Ontology sizes shown in the Ontology Atlas coverage view for the three ontologies used in the experiments.

5 Adaptive Personalization with Bayesian Knowledge Tracing

The tutor maintains a separate BKT model for each KC. After each question response, the tutor updates the KC-specific mastery estimate $\hat{m}_{i,k}$ using standard BKT parameters for initial mastery, learning, guess, and slip (Corbett and Anderson, 1994). These estimates drive reading adaptation but do not directly modify hidden learner state.

We use BKT as the tutor-visible observation model, not as the cognitive model of reading. Passages update hidden semantic state; quiz responses provide noisy evidence for KC mastery estimates. BKT is useful here because it gives the adaptation policy an interpretable KC-level state, so we report both hidden-state summaries and tutor-side BKT gain.

The adaptation policy maps low estimated mastery to more supportive readings. In the adaptive condition, low $\hat{m}_{i,k}$ can increase explanatory depth, example density, explicit misconception refutation, repeated weak-KC coverage, and reader–text matching. Higher $\hat{m}_{i,k}$ yields leaner variants. The control condition uses the same ontology, source corpus, and assessment machinery, but holds the reading configuration fixed rather than conditioning it on $\hat{m}_{i,k}$.

6 Experimental Setup

We ran fixed-iteration experiments in three subject ontologies: foundations of computer science, general biology, and introductory inorganic chemistry. Table 2 summarizes the ontology sizes surfaced in Ontology Atlas for those three subjects.

For each subject, we sampled four LOs across two chapters, created 50 matched simulated learners per condition, ran three reading–assessment cycles per LO, and generated three KC-aligned multiple-choice items per cycle. This yields 1,800 item-level responses per condition per subject.

Adaptive and control runs were matched at initialization: each control run reused the same learner

seeds and initial hidden states as its adaptive counterpart. Assessment delivery context was set to `summative`, so quiz answers updated tutor-visible BKT state but did not directly change hidden semantic knowledge. We report observed accuracy, tutor-side BKT gain, hidden mastery gain, and misconception reduction. Because learner-level paired trajectories are preserved only for accuracy and BKT gain, inferential tests are limited to those outcomes.

7 Results

7.1 Subject-level outcomes

Figure 4 and Table 3 summarize adaptive-minus-control deltas. Computer science shows the clearest benefit: accuracy improved from 0.834 to 0.865 ($\Delta = +0.031$, 95% CI $[+0.004, +0.058]$, $p = 0.026$), and learner-level BKT gain improved from 0.416 to 0.454 ($\Delta = +0.038$, 95% CI $[+0.002, +0.074]$, $p = 0.037$). Chemistry was positive but inconclusive (accuracy $\Delta = +0.019$, $p = 0.104$; BKT gain $\Delta = +0.009$, $p = 0.540$). Biology was neutral to slightly negative (accuracy $\Delta = -0.005$, $p = 0.691$; BKT gain $\Delta = -0.014$, $p = 0.468$). Descriptive hidden-state endpoints in Table 4 favored adaptation for hidden mastery in all three subjects and for misconception reduction in computer science and chemistry.

Subject	Cycle	Observed Accuracy			Tutor-side BKT gain		
		Adaptive	Control	Δ	Adaptive	Control	Δ
Computer Science	1	0.802	0.760	+0.042	0.180	0.157	+0.024
	2	0.890	0.853	+0.038	0.168	0.146	+0.023
	3	0.904	0.890	+0.014	0.106	0.114	-0.008
General Biology	1	0.794	0.795	-0.001	0.174	0.167	+0.007
	2	0.864	0.849	+0.015	0.122	0.111	+0.011
	3	0.877	0.906	-0.029	0.073	0.106	-0.033
Inorganic Chemistry	1	0.807	0.797	+0.010	0.190	0.190	+0.000
	2	0.868	0.828	+0.040	0.146	0.136	+0.010
	3	0.885	0.878	+0.007	0.092	0.092	+0.000

Table 3: Subject-level adaptive versus control outcomes.

7.2 Cycle-level trajectories

Figure 5 shows that adaptive reading led control in every computer-science and chemistry cycle, with the largest chemistry margin in cycle 2. Biology showed a crossover pattern: nearly tied in cycle 1, adaptive ahead in cycle 2, and control ahead in cycle 3.

8 Discussion

The results suggest domain-dependent effects of the adaptive policy. Computer science provides the most consistent evidence of benefit, chemistry

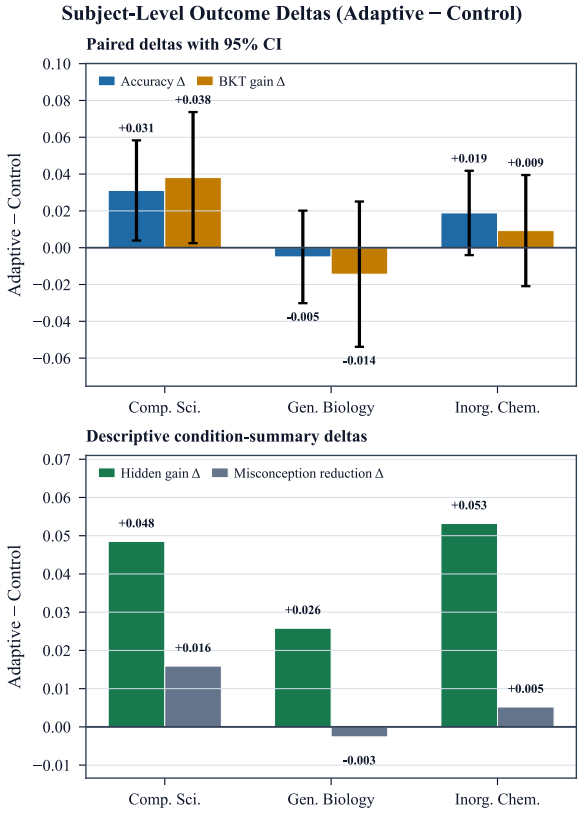

Figure 4: Subject-level adaptive-minus-control deltas.

Subject	Hidden mastery gain			Misconception reduction		
	Adapt.	Ctrl.	Δ	Adapt.	Ctrl.	Δ
Computer Science	0.521	0.473	+0.049	0.072	0.056	+0.016
General Biology	0.388	0.362	+0.026	0.045	0.048	-0.003
Inorganic Chemistry	0.509	0.456	+0.053	0.061	0.056	+0.005

Table 4: Descriptive hidden-state outcomes from saved experiment summaries.

shows smaller positive but inconclusive gains, and biology shows a mixed pattern in which hidden mastery improved descriptively without improving observed accuracy or tutor-side BKT gain.

This divergence between hidden and observed outcomes is informative. In the simulator, answers depend on hidden KC knowledge, episodic traces, misconception weight, item difficulty, attention stability, and distractor overlap. A reading can therefore improve hidden knowledge while producing little observed gain if retrieval is brittle, items are noisy, or distractors remain competitive.

The results also illustrate the role of Ontology Atlas in auditing ontology granularity and coverage. Adaptive benefit varied by LO and subject, so ontology quality and KC granularity are not merely upstream bookkeeping issues. If an LO is too broad, if its KCs are diffuse, or if retrieved

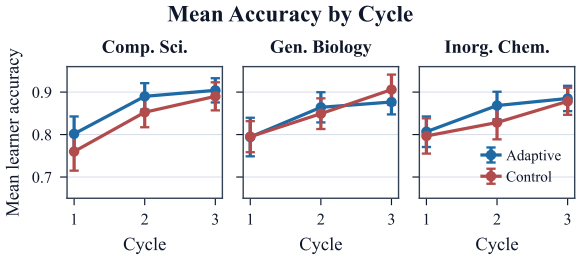

Figure 5: Mean learner-level quiz accuracy by fixed iteration cycle.

evidence is weakly aligned with target KCs, both personalization and evaluation become harder to interpret.

9 Conclusion

We presented a full-stack offline framework for evaluating adaptive personalization of educational readings with theory-grounded simulated learners. The framework links ontology-grounded content generation, KC-level BKT mastery tracking, Ontology Atlas ontology authoring, CI-based comprehension, DIME-style reader heterogeneity, KREC-style misconception revision, and score-based answer selection over explicit learner memory. Across three sampled subject ontologies, adaptive reading yielded reliable gains in computer science, smaller inconclusive gains in chemistry, and neutral to slightly negative observed outcomes in biology.

Limitations

The framework has several limitations. First, simulated learners are not substitutes for real students. The simulator is designed for inspectable offline policy screening, not as a validated estimator of classroom effect sizes, and it has not yet been calibrated against human response distributions, learning curves, distractor choices, misconception persistence, or subgroup heterogeneity.

Second, the modeled instructional setting is intentionally narrow: learners read independently, assessments provide observations but no direct instruction, and peer interaction, instructor intervention, classroom discussion, feedback, and retrieval-practice effects are outside the simulation. This isolates reading as the intervention, but it does not approximate courses where instructors diagnose errors, peers explain concepts, or quizzes teach through feedback and retrieval practice.

Third, the ontology and labeling pipeline are central sources of uncertainty. The ontologies are corpus-grounded and built from the same Wikibooks textbooks used for retrieval and generation. Errors in LO/KC granularity, chunk labels, retrieved evidence, or item alignment can propagate through generation, adaptation, and evaluation, so cross-domain differences may reflect content-pipeline quality as well as the adaptive policy. Porting to a new course therefore still requires domain review of the ontology, labels, generated readings, and assessment items rather than only rerunning the automatic pipeline.

Fourth, the simulator uses simplified operational heuristics: clarity is approximated from length, refutation strength from lexical cues, and the encoding and response equations use fixed coefficients. Sensitivity to coefficient perturbations, alternative detectors, and ablations remains untested.

Fifth, the readability signal is deliberately shallow. New Dale–Chall provides an open, reproducible proxy based on word familiarity and sentence length (Chall and Dale, 1995). It was used because the learner model includes vocabulary and background-knowledge factors: Flesch Reading Ease is driven by average sentence length and average syllables per word (Flesch, 1948), whereas New Dale–Chall incorporates word familiarity in addition to sentence length. This makes New Dale–Chall a closer operational proxy for whether a passage contains vocabulary likely to be unfamiliar to a reader, including short domain terms that syllable-based formulas may not penalize. However, it does not model discourse cohesion, diagrams, mathematical notation, domain-specific prior vocabulary, or whether a learner already knows terms such as *ATP*, *loop*, or *ion*. Future work should compare New Dale–Chall with Flesch-Kincaid, domain-vocabulary measures, and learned reader–text difficulty models.

Sixth, score-based answer selection supports bounded competence and auditability, but it may under-represent open-ended reasoning, explanation quality, and strategic test-taking. We also do not compare against constrained LLM-based student simulators or other cognitive simulators, and preprocessing artifacts may depend on the embedding and instruction models used.

Finally, the saved result format preserves learner-level paired trajectories for accuracy and BKT gain, but only condition-level summaries for hidden mastery and misconception endpoints. Future runs should preserve learner-level hidden-state trajecto-

ries for paired analysis.

Ethics Statement

The system uses open educational content and simulated learners. Any deployment decisions affecting students require human-subject validation under appropriate institutional review and privacy safeguards.

References

Shivam Bansal and Chaitanya Aggarwal. 2026. textstat: Calculate statistical features from text. Python package, version 0.7.13. Accessed 2026-03-30.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability Revisited: The New Dale–Chall Readability Formula*. Brookline Books, Cambridge, MA.

Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

Jennifer G. Cromley and Roger Azevedo. 2007. Testing and refining the direct and inferential mediation model of reading comprehension. *Journal of Educational Psychology*, 99(2):311–325.

Jennifer G. Cromley, Lindsey E. Snyder-Hogan, and Ulana A. Luciw-Dubas. 2010. Reading comprehension of scientific text: A domain-specific test of the direct and inferential mediation model of reading comprehension. *Journal of Educational Psychology*, 102(3):687–700.

Xinyi Ding and Eric C. Larson. 2021. On the interpretability of deep learning based models for knowledge tracing. *Preprint*, arXiv:2101.11335.

Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Mark J. Gierl, Okan Bulut, Qi Guo, and Xinxin Zhang. 2017. Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of Educational Research*, 87(6):1082–1116.

Xiaoli Huang, Wei Xu, and Ruijia Liu. 2025. Effects of intelligent tutoring systems on educational outcomes: Evidence from a comprehensive analysis. *International Journal of Distance Education Technologies*, 23(1):1–25.

Tanja Käser and Giora Alexandron. 2024. Simulated learners in educational technology: A systematic literature review and a turing-like test. *International Journal of Artificial Intelligence in Education*, 34(2):545–585.

Panayiota Kendeou. 2024. A theory of knowledge revision: The development of the krec framework. *Educational Psychology Review*, 36(2):44.

Panayiota Kendeou and Edward J. O'Brien. 2014. The knowledge revision components (krec) framework: Processes and mechanisms. In David N. Rapp and Jason L. G. Braasch, editors, *Processing Inaccurate Information: Theoretical and Applied Perspectives from Cognitive Science and the Educational Sciences*, pages 353–377. MIT Press, Cambridge, MA.

Walter Kintsch. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163–182.

Benoît Lemaire, Guy Denhière, Cédrick Bellissens, and Sandra Jhean-Larose. 2006. A computational model for simulating text comprehension. *Behavior Research Methods*, 38(4):628–637.

Angélique Létourneau, Marion Deslandes Martineau, Patrick Charland, John Alexander Karran, Jared Boasen, and Pierre-Majorique Léger. 2025. A systematic review of AI-driven intelligent tutoring systems (ITS) in K-12 education. *npj Science of Learning*, 10(1):29.

Paul van den Broek and Panayiota Kendeou. 2008. Cognitive processes in comprehension of science texts: The role of co-activation in confronting misconceptions. *Applied Cognitive Psychology*, 22(3):335–351.

Kurt VanLehn, Stellan Ohlsson, and Rod Nason. 1994. Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education*, 5(2):135–175.

Wikibooks contributors. 2026a. Foundations of Computer Science. https://en.wikibooks.org/wiki/Foundations_of_Computer_Science. Accessed 2026-05-13.

Wikibooks contributors. 2026b. General Biology. https://en.wikibooks.org/wiki/General_Biology. Accessed 2026-05-13.

Wikibooks contributors. 2026c. Introduction to Inorganic Chemistry. https://en.wikibooks.org/wiki/Introduction_to_Inorganic_Chemistry. Accessed 2026-05-13.

Zhihao Yuan, Yunze Xiao, Ming Li, Weihao Xuan, Richard Tong, Mona T. Diab, and Tom Mitchell. 2026. Towards valid student simulation with large language models. *Preprint*, arXiv:2601.05473.