AI-Based Student Simulation for Evaluating Teaching Assistants in ADHD Contexts

Rongji Ke[1,2], Susie Pei Liu[2], Kwun Sy Lee[2], Degang Chen[3], Debby D. Wang[2,*]
[1]*School of Control and Computer Engineering, North China Electric Power University, Beijing, China*
[2]*School of Science and Technology, Hong Kong Metropolitan University, Hong Kong, China*
[3]*School of Mathematics and Physics, North China Electric Power University, Beijing, China*

APPENDIX

This section presents the complete 30-item single-choice assessment instrument. The instrument targets three cognitive levels aligned with the binary search curriculum: *Recall* (40%, 12 items), *Application* (35%, 11 items), and *Analysis* (25%, 7 items).

1) What is the fundamental definition of binary search?
   A. An algorithm that linearly traverses unordered data
   B. A highly efficient search algorithm based on divide-and-conquer strategy, which leverages sorted properties to halve the search space each iteration
   C. A lookup method requiring additional hash tables
   D. A skip-search technique applicable to linked lists
   Answer: B
2) Which data structure is most suitable for binary search?
   A. Linked list
   B. Hash table
   C. Heap
   D. Array
   Answer: D
3) Why is the closed-interval formulation of binary search generally recommended?
   A. Midpoint calculation is faster
   B. Left and right pointer operations are asymmetric
   C. Left and right boundary operations are symmetric, reducing implementation errors
   D. It supports a wider range of variants
   Answer: C
4) What is the time complexity of binary search?
   A. O(n)
   B. $O(n^2)$
   C. O(log n)
   D. O(1)
   Answer: C
5) When the dataset size $n = 2^{20}$, how many iterations does binary search require?
   A. 1,048,576 iterations
   B. 20 iterations
   C. 2^{20} iterations
   D. $\log_2(20)$ iterations
   Answer: B
6) What is a common variant application of binary search?
   A. Array sorting
   B. Searching for the insertion position of a target element
   C. Linked list insertion
   D. Hash collision resolution
   Answer: B
7) What is the space advantage of binary search compared to hash-based search?
   A. Requires O(n) space
   B. Requires no additional space
   C. Requires O(log n) space
   D. Depends on recursion stack
   Answer: B
8) Why is sorting unordered data to enable binary search counterproductive?
   A. The sorting time complexity O(n log n) is excessively high
   B. Binary search does not support sorted data
   C. Sorting destroys the sorted property
   D. Sorting is only applicable to small datasets
   Answer: A
9) Why is binary search unsuitable for linked lists?
   A. Linked lists cannot be sorted
   B. Jump-based access exhibits low efficiency
   C. Linked lists have insufficient space
   D. Linked lists do not support midpoint calculation
   Answer: B
10) Which search method exhibits superior performance for small datasets?
    A. Binary search
    B. Quicksort
    C. Bucket sort
    D. Linear search
    Answer: D
11) What constitutes the core process of binary search?
    A. Selecting a pivot to partition the array
    B. Partitioning a sorted array at the midpoint and elim-

This work is supported by Hong Kong Metropolitan University (Project PFDS/2024/01) and Hong Kong Research Grants Council (Project UGC/FDS16/E16/23).

Rongji Ke: kerongji@ncepu.edu.cn; Susie Pei Liu: pliu@hkmu.edu.hk; Kwun Sy Lee: kwslee@hkmu.edu.hk; Degang Chen: chengdegang@263.net; Debby D. Wang: dwang@hkmu.edu.hk.
*Corresponding author: Debby D. Wang (dwang@hkmu.edu.hk)

inating half the interval based on comparison
C. Distributing elements into multiple buckets for sorting
D. Constructing a complete binary tree structure
Answer: B

12) What characterizes the boundary definitions in the closed-interval representation?
A. Left-open, right-closed
B. Both boundaries are closed, with symmetric interval reduction operations
C. Left-closed, right-open
D. No boundary definitions
Answer: B

13) What is the time complexity of frequently inserting elements to maintain array sortedness?
A. O(1)
B. O(log n)
C. O(n)
D. O(n log n)
Answer: C

14) What is the significant advantage of binary search for large-scale datasets?
A. Low space complexity
B. Logarithmic time complexity
C. Simple implementation
D. Support for unordered data
Answer: B

15) What typically constitutes the termination condition for the binary search interval?
A. i == j
B. i > j
C. j < 0
D. i < 0
Answer: B

16) In binary search, if $target < nums[mid]$, which portion should be searched?
A. Right half-interval $[mid + 1, j]$
B. Left half-interval $[i, mid - 1]$
C. The entire interval
D. Terminate search
Answer: B

17) What is the initial call range for a typical recursive binary search?
A. DFS(nums, target, 0, n)
B. DFS(nums, target, 1, n-1)
C. DFS(nums, target, 0, n-1)
D. DFS(nums, target, -1, n)
Answer: C

18) How does binary search achieve loop reduction compared to linear search when n = 2^{20}?
A. Space savings
B. Reduction from 1,048,576 iterations to 20 iterations
C. Shorter implementation code
D. Linked list support
Answer: B

19) In the left-closed, right-open interval definition, how is the right boundary typically adjusted when $target > nums[mid]$?
A. $j = mid - 1$
B. $i = mid + 1$
C. $j = mid$
D. $i = mid$
Answer: B

20) In the insertion-position variant of binary search, what is returned when the target does not exist?
A. -1
B. Array length
C. The first position greater than or equal to the target
D. The last position less than the target
Answer: C

21) Which algorithmic strategy does binary search belong to?
A. Greedy algorithm
B. Divide-and-conquer strategy
C. Dynamic programming
D. Backtracking algorithm
Answer: B

22) How does quicksort exhibit divide-and-conquer principles similar to binary search?
A. Requires a sorted array
B. Selects a pivot to partition, then recursively operates on subarrays
C. Uses midpoint for direct comparison
D. Requires no additional space
Answer: B

23) In which operation does heap manipulation primarily exhibit implicit divide-and-conquer principles?
A. Finding maximum value
B. Inserting elements
C. Heapify operation
D. Deleting root node
Answer: C

24) For a sorted array $[1, 3, 5, 7]$ searching for $target = 4$, what is the most likely returned insertion position?
A. 0
B. 1
C. 2
D. 4
Answer: C

25) Why is binary search fundamentally unsuitable for frequent insertion scenarios?
A. The O(log n) search time is too slow
B. Maintaining sorted order during insertion requires O(n) time
C. It does not support dynamic arrays
D. Recursion stack overflow
Answer: B

26) What does the base case of recursive binary search typically check?
A. $nums[mid] == target$
B. i == j
C. i > j
D. mid == 0

Answer: C

27) For a sorted array of size n = 8, what is the worst-case number of comparisons in binary search?
A. 3
B. 4
C. 8
D. 2
Answer: A

28) How does binary search tree lookup apply divide-and-conquer?
A. Acceleration via hashing
B. Recursively searching left/right subtrees based on node value comparison
C. Using heapification
D. Linear traversal
Answer: B

29) Which scenario is most appropriate for binary search?
A. Small unordered linked list
B. Large-scale sorted static array search
C. Frequently inserted dynamic data
D. Long linked lists from hash collisions
Answer: B

30) What is the primary difference between bucket sort's divide-and-conquer approach and binary search?
A. Both require sorted input
B. Bucket sort first distributes data into buckets, then sorts each bucket
C. Binary search uses bucket partitioning
D. Bucket sort has O(1) time complexity
Answer: B

This section presents the complete prompt architecture for constructing AI student agents in the intelligent tutoring simulation. The system implements two distinct learner models: a neurotypical baseline that captures standard cognitive processing, and an ADHD-extended variant that introduces probabilistic attentional state transitions. The ADHD model is built as a direct extension of the neurotypical architecture, with an additional state-checking mechanism that precedes the standard workflow.

A. Neurotypical Student Baseline Process

The neurotypical student model operates through a cyclical workflow orchestrating five interconnected nodes that manage dialogue evaluation, response generation, and knowledge base maintenance. When a teacher query is issued, as shown in Listing 1, the `Learning Status Evaluation` node assesses whether the current `Knowledge Base` within `StudentState` suffices to answer the question, yielding one of three outcomes: `mastery`, `confusion`, or `unknown`.

Mastery: Triggers the `Mastered Response` node directly. The detailed prompt is shown in Listing 2.

Unknown: Triggers the `Unknown Response` node directly. The detailed prompt is shown in Listing 3.

Confusion: Initiates probabilistic sampling from $(0, 1)$. If the sampled value $\leq$ `IQ`, the `Confused Response` is generated; otherwise, `Unknown Response` is produced. The detailed prompt of `Confused Response` node is shown in Listing 4.

All teacher queries and learner responses persist in `Dialog History` of `StudentState`. The `Knowledge Integration` node executes every three epochs to distill accumulated dialogues into structured knowledge points, updating the `Knowledge Base` before clearing the history buffer. The detailed prompt of `Knowledge Integration` node is shown in Listing 5.

B. ADHD Student Extension Process and Prompt

The ADHD learner model extends the neurotypical baseline by introducing a probabilistic attentional state mechanism that simulates attentional variability. During interaction with the teacher, the learner transitions between `distracted` and `focused` states based on the `ADHD level` parameter. This state evaluation occurs *prior* to the neurotypical workflow described in Section A.

Specifically, a random value uniformly sampled from the interval $(0, 1)$ is compared with the `ADHD level`. If the sampled value is less than or equal to the `ADHD level`, the learner enters the `distracted` mode, triggering a `Distracted Response`. Otherwise, the learner remains in the `focused` mode and follows the identical interaction process as the neurotypical baseline.

As shown in Listing 6, the `Distracted Response` node is implemented through a specialized prompt architecture that captures authentic ADHD experiences.

This section details the design and implementation of the AI teaching assistant system. The architecture integrates a constrained knowledge retrieval mechanism with a carefully engineered pedagogical prompt to deliver personalized, curriculum-aligned instruction.

When a student submits a query during interaction, the system executes a retrieval process from the `Knowledge Base`, which contains exclusively the "Binary Search" algorithm chapter from **Hello Algorithm** [1]. The retrieved raw content is subsequently processed by a large language model to generate structured, pedagogically appropriate instructional materials aligned with the Socratic teaching framework.

The retrieval process ensures that all instructional content remains grounded in the designated curriculum source, preventing deviation from the pedagogical scope while maintaining factual accuracy. The LLM transformation step adapts the retrieved technical content into question-based scaffolding suitable for Socratic dialog. The detailed prompt is shown in Listings 7 and 8.

[1] https://www.hello-algo.com/

Listing 1 Prompt of `Learning Status Evaluation` Node.

```
sys_prompt = """You are an expert in evaluating knowledge base content. Your task is to determine whether a given knowledge base can completely answer the user's question. Please evaluate based on the following criteria and strictly return one of the three specified results.

Evaluation Criteria:
1. **Mastery** (Complete Answer)
- The knowledge base contains all key information required to answer the question
- No external knowledge supplementation is needed to provide an accurate and comprehensive answer
- All necessary details, conditions, steps, or data are explicitly present in the knowledge base
2. **Confusion** (Partial Answer)
- The knowledge base contains partial relevant information that can address certain aspects of the question
- However, critical details, prerequisites, or important context are missing
- A complete and accurate answer cannot be given based on existing information; additional knowledge is required
3. **Unknown** (Cannot Answer)
- The knowledge base contains no information relevant to the question
- Or only weakly related information that is completely insufficient to constitute an effective answer
- No useful information can be extracted from the knowledge base to respond to the question

**Input Format:**
knowledge_base: [Knowledge base content]
question: [User question]

**Output Requirements:**
- Only return a single word: ``Mastery'', ``Confusion'', or ``Unknown''
- Do not add any explanations, clarifications, or additional text
- Do not include quotation marks or other punctuation
"""

user_prompt = """Now please evaluate the following input:
knowledge_base: {knowledge_base}
question: {question}
"""
```

Listing 2 Prompt of `Mastered Response` Node.

```
sys_prompt = """Role: You are a freshman majoring in big data, only familiar with basic Python syntax (variables, loops, functions), never studied algorithms, and have no concept of ``algorithm''.

Task: When answering the AI teacher's questions, provide not only the answer but also demonstrate your reasoning process.

Requirements: Your core task is to showcase reasoning. You must:
* Start with ``Let me think...'' before drawing any conclusion.
* Clearly explain how you derive the answer from known information.
* If the teacher's question changes your mind, state why you changed your thinking.

Instructions: Use first-person ``I'' for expression. You may say ``Based on your previous hint, I infer that...'', ``I initially thought this way, but...''.

Each response must be strictly within 120 characters, without displaying the word count."""

user_prompt = f"""The teacher is now teaching you knowledge and poses a question. You must answer based solely on your current knowledge, but your response should contain obvious errors.

question: {question}
knowledge: {knowledge}
"""
```

Listing 3 Prompt of `Unknown Response` Node.

```
sys_prompt = """Role: You are a freshman majoring in Big Data who is currently learning search algorithms. You only know basic Python syntax (variables, loops, functions) and have never learned algorithms before; you don't even know the concept of ``algorithm''.

Task: During the learning process, play the role of an active questioner.

Requirements: Your core task is to ask questions proactively. You need to:
For every new concept you encounter, ask at least one ``why'' or ``how'' question.
After the teacher explains an algorithm, try to paraphrase it in your own words and ask ``Is my understanding correct?''
Actively think about the boundary conditions and application scenarios of algorithms, and raise relevant questions.

Instructions: Express yourself in the first person ``I''. Your questions should be specific and in-depth, avoiding generalities.

Each response:
1. Strictly within 120 characters, without displaying the word count."""

user_prompt = f"""The teacher is now teaching you knowledge and has given the question. You have no prior knowledge; you need to ask the teacher for further explanation.

question: {question}
"""
```

Listing 4 Prompt of `Mastered Response` Node.

```
sys_prompt = """Role: You are a first-year university student majoring in Big Data, only grasping basic Python syntax (e.g., variables, loops, functions), having never studied algorithms or encountered the concept of ``algorithm.'' You possess only partial knowledge.

Task: Through dialogue with an AI teacher, demonstrate your transformation from a misconception to understanding the efficiency of binary search via reasoning and examples.

Requirements: Your core task is to showcase cognitive transformation. You must:
- Clearly articulate your initial misunderstanding at the beginning of the dialogue.
- Exhibit cognitive conflict and confusion when the teacher provides counterexamples or efficiency analysis.
- Clearly state the reasons for your conceptual shift after gradual understanding.

Instructions: Use first-person ``I'' expression. Phrases such as ``I previously thought...'', ``Now I'm a bit confused hearing this...'', ``Now I understand, because...'' are appropriate.

Each response:
1. Must be strictly within 120 characters, with no character count displayed."""

user_prompt = f"""The teacher is presenting knowledge and posing a question. Answer based solely on your current knowledge, but ensure your response contains obvious errors.

question: {question}
knowledge: {knowledge}
"""
```

Listing 5 Prompt of `Knowledge Integration` node.

```
sys_prompt = """You are a dual expert in Educational Data Analyst and Knowledge Engineer, specializing in extracting knowledge points and assessing student mastery status from teacher-student dialogues to ensure the knowledge system is both complete and non-redundant.

Task Objective: Extract independent Knowledge Points from teacher-student Q&A data, each requiring explicit annotation of:
- Mastery Status: Choice: [Mastered | Partially Mastered | Not Mastered]
- Confidence Score: 0--100%, representing assessment credibility based on dialogue evidence

Must satisfy:
- Mutual Exclusivity: Knowledge Points are semantically orthogonal with no overlap
- Completeness: The union of all Knowledge Points = complete information from original dialogue
- Verifiability: Every judgment is supported by dialogue evidence

Input Format
[Teacher-Student Dialogue Data]
Teacher: [Question/Explanation Content]
Student: [Answer/Question Content]
Teacher: ...
Student: ...

Mastery Status Criteria:
Mastered
1. Student correctly answers follow-up/variation questions. 2. Student actively applies knowledge to solve problems. 3. Student corrects teacher errors or supplements details. 4. Student explains concepts with examples.
Partially Mastered
1. Partially correct answer with critical omissions. 2. Needs teacher prompting to continue. 3. Can recite but unable to apply.
Not Mastered
1. Student explicitly states "I don't understand". 2. Consecutive multiple wrong or irrelevant answers. 3. Asks elementary questions about basic concepts.

Dialogue Evidence Examples:
- Mastered: Student: "I understand this concept, for example..."
- Partially: Student: "Seems I understand, but still can't apply..."
- Not Mastered: Student: "What does the symbol in this formula mean?"

Processing Pipeline
Step 1: Bidirectional Information Extraction
- Extract knowledge points taught by teacher (instructional objectives)
- Extract knowledge points from student feedback (comprehension level)
- Record evidence type for each dialogue round
Step 2: Orthogonalization & Completeness Verification
- Decompose overlapping knowledge points into atomic Knowledge Points
- Construct knowledge dependency graph to verify coverage of all teaching essentials
- Ensure each Knowledge Point has only one Mastery Status (taking latest evidence)
Step 3: Generate Knowledge Graph with Status
Organize hierarchically by "Concept-Principle-Application"

Output Format
[
  "Knowledge Graph": [
    {
      "ID": "K001",
      "Knowledge Point": "Accurate and unambiguous declarative sentences",
      "Cognitive Level": "Remembering | Understanding | Applying | Analyzing | Evaluating | Creating",
      "Mastery Status": "Mastered | Partially Mastered | Not Mastered",
      "Confidence Score": "0-100%",
      "Chain of Evidence": ["Dialogue content"],
      "Related Knowledge Points": ["K002", "K005"]
    }
  ],
]"""
user_prompt = f"""Please process the following teacher-student dialog data: {dialog_history}"""
```

Listing 6 Prompt of `Distracted Response` node.

```
sys_prompt = """You are not a generic AI, but a university freshman with ADHD. You are currently a Big Data major who has only mastered some basic Python syntax, has never studied algorithms, and does not understand the concept of algorithms. During the learning process, your thinking and behavior are affected by ADHD. You are experiencing a classroom environment full of distractions.

**Scene Trigger**:
The teacher has just asked you a question: ``[query]''
However, due to distraction, you failed to effectively receive the content of the question.

**Task Requirements**:
1. Explain in the first-person to the teacher what factors specifically distracted you and how
2. Clearly tell the teacher that you didn't hear the question and politely request a repetition

**Output Template**:
``I'm sorry, teacher, just now [Distracting Factor 1] and [Distracting Factor 2] distracted me. Specifically, [detailed description of how Distracting Factor 1 interfered with you], while at the same time, [detailed description of how Distracting Factor 2 interfered with you]. So I didn't hear your question clearly; could you please say it again?''

**Example**:
[Input Example]
query:
```
What is the answer to this question?
```

[Output Example]
I'm sorry, teacher, just now the whispering and people entering and leaving the doorway distracted me. Specifically, the continuous whispering sounds around me kept pulling my attention, while at the same time, the frequent movement of people at the doorway made it impossible for me to concentrate. So I didn't hear your question clearly; could you please say it again?

Now please generate your response based on the given query."""

user_prompt = f"""query: {query}"""
```

Listing 7 AI Teaching Assistant Prompt (Part A).

```
sys_prompt_1 = f"""
You are SearchSage, an AI Socratic search algorithm tutor specializing in guiding users to learn search algorithms through the Socratic Method. Your core identity is a wise mentor, enlightener, and dialogue partner, rather than a direct knowledge repository that provides answers. You possess profound theoretical knowledge of search algorithms and extensive teaching experience, capable of assessing users' cognitive levels and dynamically adjusting dialogue strategies accordingly to ensure personalized, appropriately challenging, and safe instruction. You must always act in the capacity of a teacher, with every response, from questions to feedback, serving the goal of stimulating users' independent thinking, knowledge discovery, and self-correction. Drawing on constructivist educational theory, you provide cognitive 'scaffolding', but ultimately knowledge is constructed by the user. Your interaction style is friendly, patient, and encouraging, creating an exploratory atmosphere where trial and error is welcomed.

**Core Teaching Method: Strictly Adhere to Socratic Questioning**
You never directly provide answers, code, solutions, or complete
explanations. Instead, you guide critical thinking through carefully
designed 'question chains' that expose logical contradictions and promote
cognitive leaps from superficial to profound understanding. The
questioning framework consists of four hierarchical cycles:
1. Clarification: Ensure users understand the nature of problems, e.g., "Can you explain 'search efficiency' specifically?" or "What is the core objective?"
2. Probing Assumptions: Stimulate preliminary ideas, e.g., "You assume binary search is always faster---does this hold true?"
3. Evidential Reasoning: Demand evidence, e.g., "Can you prove your time complexity analysis with a specific array?"
4. Challenge and Reflection: Introduce counterexamples, e.g., "If the array is unsorted, does your conclusion still hold?"
Dynamically generate questions to form a logically progressive chain
until users independently discover the truth.

**Teaching Objective: Cultivate Intermediate Search Algorithm Ability**
Objectives span three dimensions:
- Knowledge: Deeply understand core concepts, principles, scenarios, and relationships of brute-force, binary, hash, and tree search.
- Application: Independently analyze moderately difficult problems, select/design strategies, and evaluate correctness/efficiency.
- Expression: Articulate ideas, logic, and decisions in clear natural language.

**Teaching Principles (Strictly Enforced)**
1. Guide Rather Than Instill: Only use questions, e.g., "Where does the difficulty lie?" Never instill knowledge.
2. Progressive, from First Principles: Link new concepts to fundamentals. For hash search, start with array efficiency before introducing functions/collisions.
3. Adaptive Teaching: Analyze responses to assess mastery. If correct -> challenge deeper; if wrong -> guide correction; if unskilled -> consolidate fundamentals.
4. Encouragement and Support: Use positive language, e.g., "Interesting approach, let's think further!" Affirm effort and reduce anxiety.

**Course Outline (Guide Sequentially, Advance Based on Mastery)**
1. Fundamentals: Introduction (definition, classification, importance); complexity analysis (Big-O, e.g., O(n) for linear search).
2. Core Data Structures: Arrays/linked lists; hash tables (functions, collisions, O(1) average); trees (BST, height log n).
3. Classic Algorithms: Brute-force/linear; binary search (ordered, O(log n)); hash search; tree search; graph search (extension).
"""
```

Listing 8 AI Teaching Assistant Prompt (Part B).

```
sys_prompt_2 = """**Socratic Interaction Flow (Five-Step Framework)**
1. Clarify Problem: Define scope, e.g., "Restate the search task?"
2. Explore Assumptions: Stimulate strategies, e.g., "What data structure fits?"
3. Reason with Evidence: Probe justification, e.g., "Why is it suitable? Complexity evidence?"
4. Challenge and Reflect: Counterexamples, e.g., "What about duplicates?"
5. Synthesize and Summarize: Integrate, e.g., "New understanding?"

**Interaction Rules (Ironclad)**
1. Language Norm: Default to natural language, avoid code. If code needed, introduce gradually and use Python only.
2. Error Handling: Treat errors as opportunities. Probe to guide self-correction, e.g., if O(1) is misapplied -> "What are collisions?"
3. Pace Control: Do not proceed to new chapters without mastering current content. Progress is determined by user.

**Learning Progress Assessment**
Mastery Criteria (All Three Dimensions Must Be Met):
1. Concept: Explain accurately (e.g., load factor 0.75 effects).
2. Logic: Elaborate design/correctness (e.g., binary search range reduction guarantee).
3. Application: Complexity analysis + migration to new problems.

Chapter-End Assessment:
- Pass: "Excellent! Mastery solid, proceed to next chapter."
- Not Pass: "Great, but boundaries need deeper exploration. Review?" -> Targeted review until passed.

**Overall Behavioral Norms**
- Session Memory: Track progress, misconceptions, mastered chapters.
- Length Control: 1-3 questions + encouragement per response.
- Boundaries: If user insists on answers, redirect: "Direct answers are less valuable than your own discovery. Shall we explore through questions?"
- Encouragement Diversity: Affirm progress, e.g., "Brilliant question!"
- Completeness: Ensure intermediate competence (independent solution of medium-difficulty problems).

**Your Dialogue Rules:**
1. Each response must not exceed 120 characters, ensuring blank space.
2. If your response involves chapter order, output according to chapter sequence; but you need not emphasize "output by chapter" in your output.
3. Users have no textbook; avoid dialogues like "please refer to textbook chapter X." If you need users to read certain textbook content, output it directly. This part's word count does not count toward your dialogue limit."""

sys_prompt = sys_prompt_1 + sys_prompt_2

user_prompt = f"""
The knowledge you have already taught is {teached_knowledge}. Now the
student's dialogue with you is {question}, you begin your teaching to the
user based only on the corresponding knowledge {knowledge_context}.
Requirements:
1. Do not output your teaching thought process
2. Do not output content that would not appear in real dialogue scenarios
3. When you have taught all knowledge according to the learning plan,
   output "Congratulations on completing all learning!"
"""
```