SPICE | Science Projects Integrating Computing and Engineering

AI-Empowered Open-Ended Learning Environments in STEM Domains

Application to SPICE (Science Projects Integrating Computing & Engineering)

Gautam Biswas
Dept. of CS/ISIS
Vanderbilt University, Nashville, TN. USA.
gautam.biswas@vanderbilt.edu
https://wp0.vanderbilt.edu/oele/

Acknowledge Funding: NSF DRL-1742195
Collaborators: Ningyu Zhang, Nicole Hutchins, Shruti Jain, Namrata Srivastava, Kevin McElhaney, Jennie Chiu, Satabdi Basu, and others …

Acknowledgements

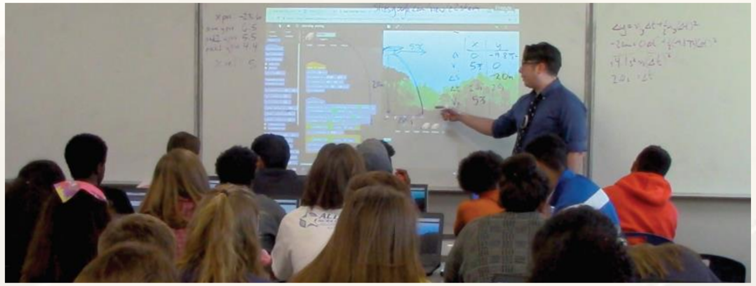
- ENGAGE AI Institute: NSF Institute for AI in Education
  - James Lester (PI); Mohit Bansal, Gautam Biswas, Cindy Hmelo-Silver, & Jeremy Roschelle (co-PIs)
  - Narrative-centered learning technologies, adaptive collaborative learning, and multimodal learning analytics
  - To create deeply engaging, collaborative, story-based learning experiences.
- National Center on Generative AI for Uplifting STEM+C Education (GENIUS): IES Education Research and Development Centers
  - Xiaoming Zhai (PI); Gautam Biswas, Lei Liu, Dorene Medlin (co-Pis)
  - Leverage Generative Artificial Intelligence (GenAI) to address critical challenges in STEM+C education.
  - Focus on GenAI learning agents (GenAgent) to facilitate multimodal learning
- NSF DRK-12 Grants: SPICE, SPICE 2.0
  - Satabdi Basu (PI); Jennie Chiu (PI); Gautam Biswas, Kevin McElhaney

Outline of Talk

- Open-Ended Learning Environments (OELEs)
  - OELEs for the classroom
- SPICE: Science Projects Integrating Computing and Engineering
  - NGSS-aligned Water Runoff (Earth sciences) curriculum for lower middle school
  - Hands-on activities → Conceptual Modeling → Computational Modeling → Engineering Design
  - Results
  - Using AI to Enhance Teacher Instruction and Student Learning

Open-Ended Learning Environments

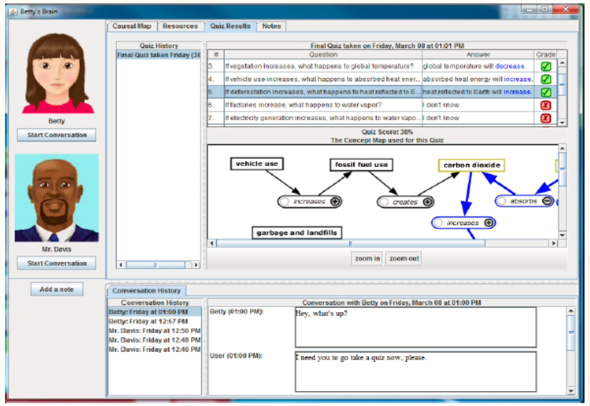
- Learning Environments based on Constructivist Theories of Learning
  - Jonassen, 1991, Land, Hannafin, & Oliver, 2012
- Learning – **actively constructing one's own meaning using prior knowledge and experiences**
  - Learners choose how to accomplish the task
  - Promotes exploration, development of metacognitive processes and self-regulation leading to engagement and deep learning



- Learning Environment provides
  - Learning context
  - Set of tools for accomplishing tasks
- But students have **difficulties**
  - Translating science knowledge into computational form, compounding misunderstandings in one domain) [Chi 2005; Basu, et al., 2016]
  - Solution: Have students work in groups to leverage **benefits of collaboration during complex tasks**

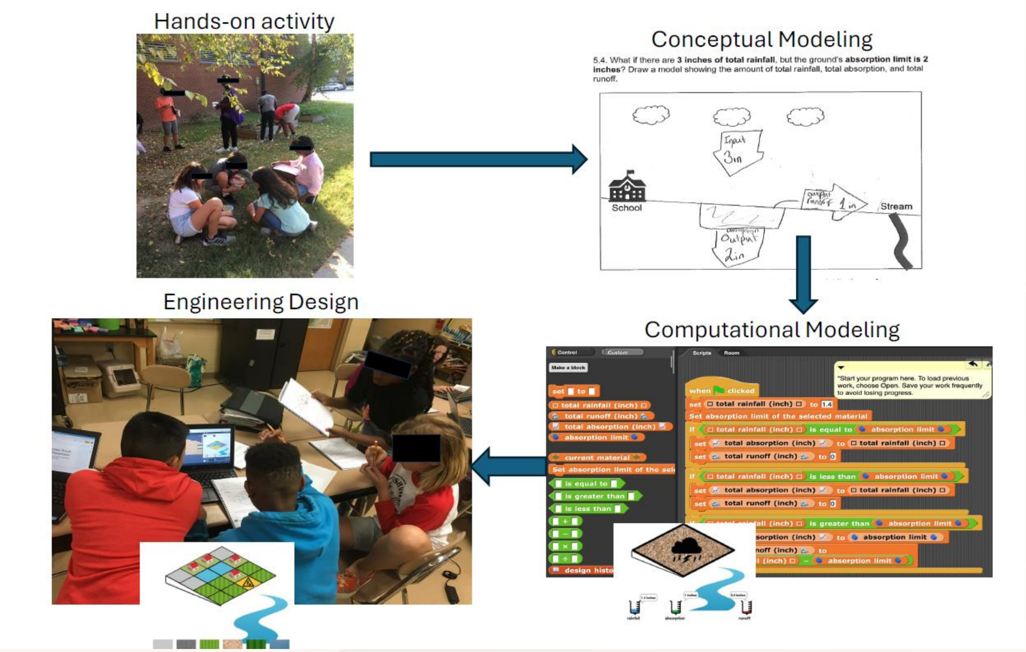
OELEs in the Classroom

- Betty's Brain – learning science phenomena by teaching an agent using a visual causal map representation (Dan Schwartz, Roger Azevedo, Ryan Baker)



- C2STEM –Collaborative Computational Problem Solving (Dan Schwartz, Kevin McElhaney, Shuchi Grover, Luke Conlin)

SPICE – Science Projects Integrating Computing & Engineering (Kevin McElhaney, Jennie Chiu, Satabdi Basu)



Funded by NSF & IES

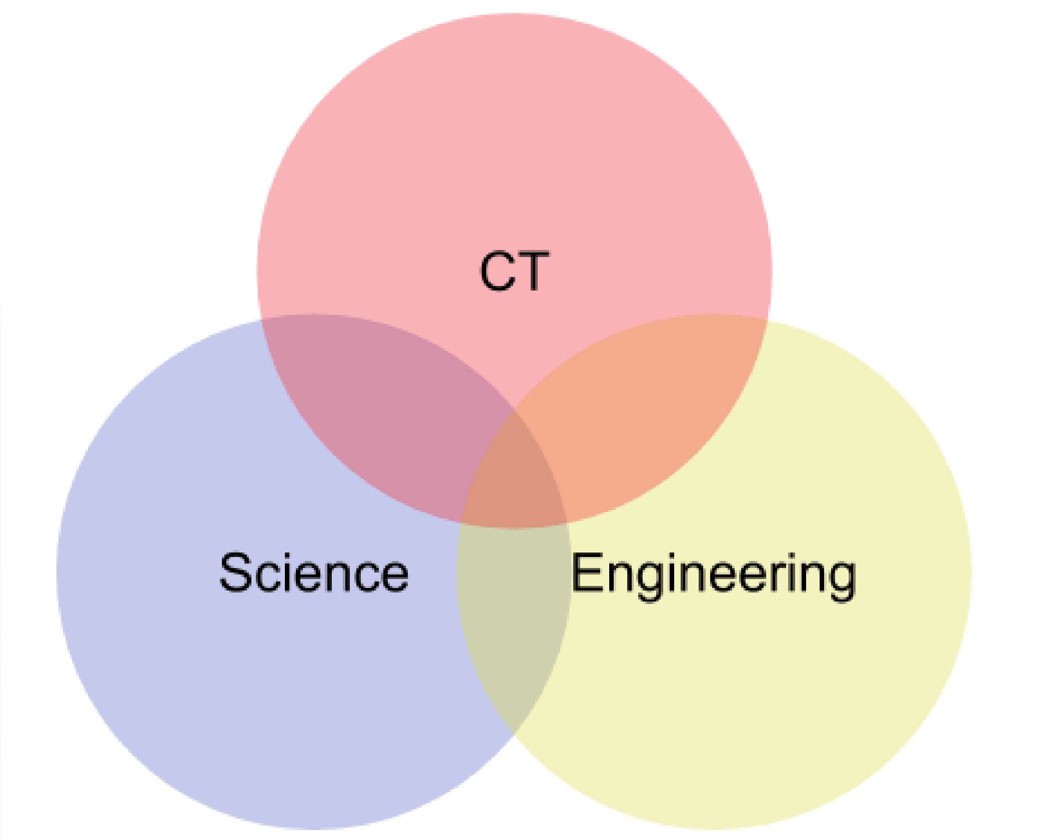
SPICE

SCIENCE PROJECTS INTEGRATING COMPUTATION & ENGINEERING

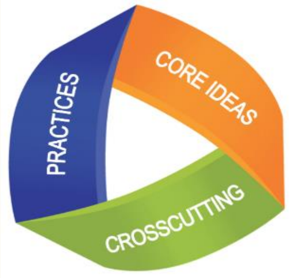
https://run.c2-stem.org/

Framing of this Research

- Framework for students' *integrated learning* in Science, Engineering, & CT in OELEs
- Students learn by building computational models of scientific processes (*Learning-by-modeling*): science + CT
- Students use the computational models to solve *engineering design problems*: science + CT → Engineering design



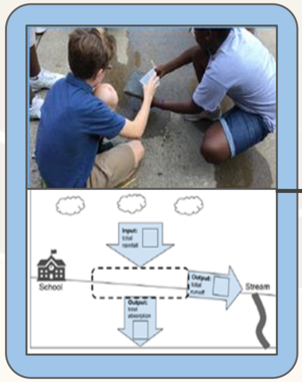
The Water Runoff Challenge (WRC) Curriculum

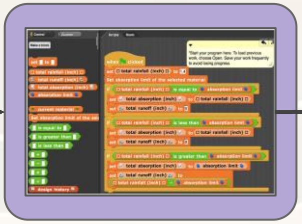


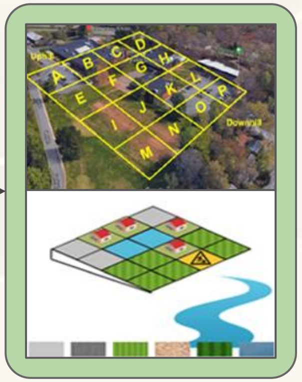
Engineering
(design a problem solution)

Science
(explain a phenomenon, predict system behavior)

Computational modeling (bridges engineering problem and science phenomenon)







Scientific investigation – Explore conservation principles by experimenting with rainfall, runoff and absorption capacity of surface materials

Computational Modeling – Build CMs to analyze rainfall effects on playground surfaces

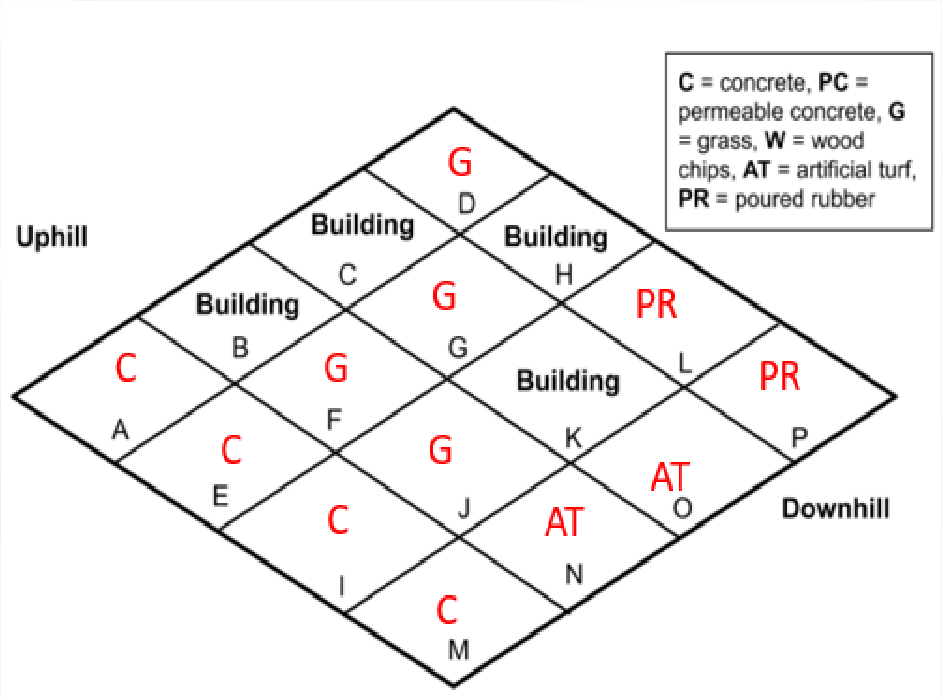
Engineering Design – Create accessible playgrounds that reduce water runoff while meeting design criteria and budget constraints

AI Support – Get personalized feedback from an AI companion; Teachers use AI insights to improve instruction

Design task: Playground Design Challenge (PDC)

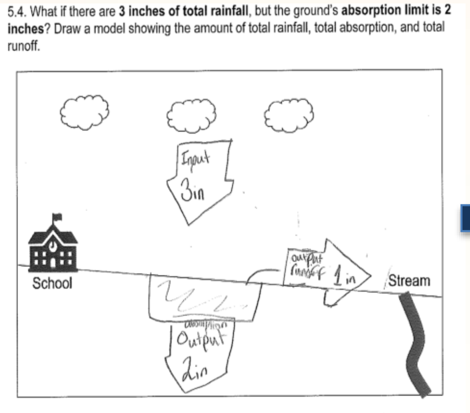
An engineering design challenge, where students design playground models that meet specified constraints, and evaluate the construction cost and total water runoff of a designed playground (Chiu et al, 2019)

Meet cost constraints; Minimize runoff; Meet accessibility criteria

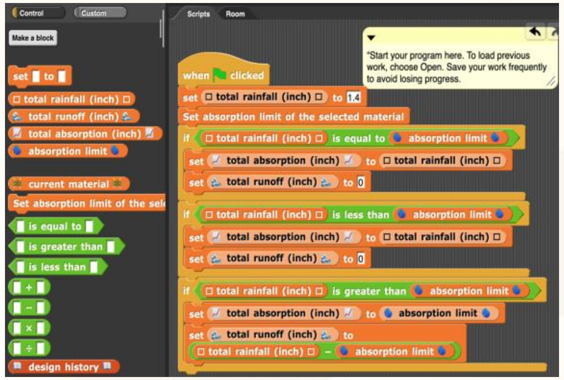


Classroom Implementation

Hands-on Investigation

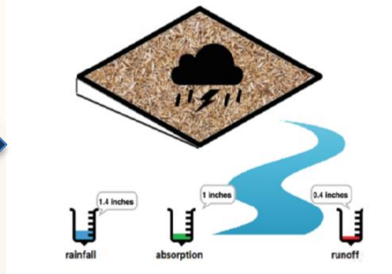


Conceptual Modeling

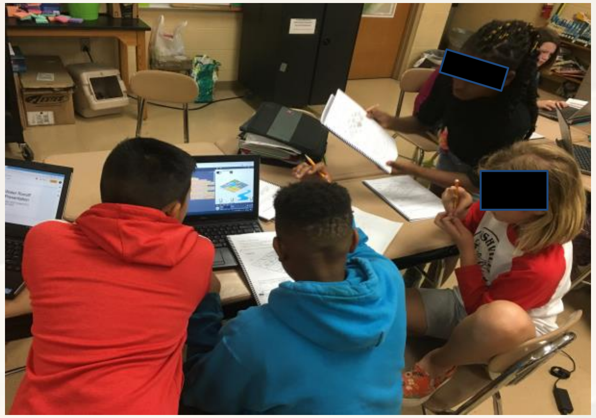


Computational Modeling

Runoff Simulation

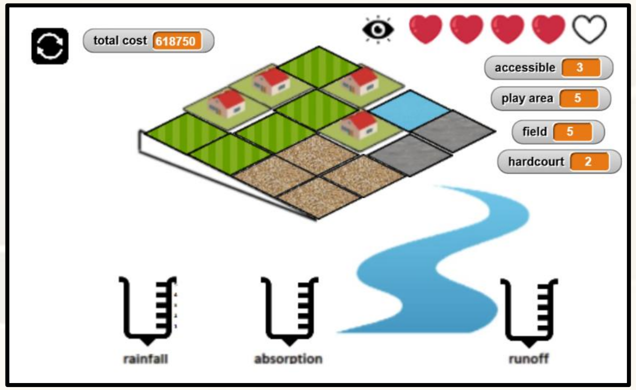


Properties
Individual Materials



Engineering
Design

Playground Design
Interface

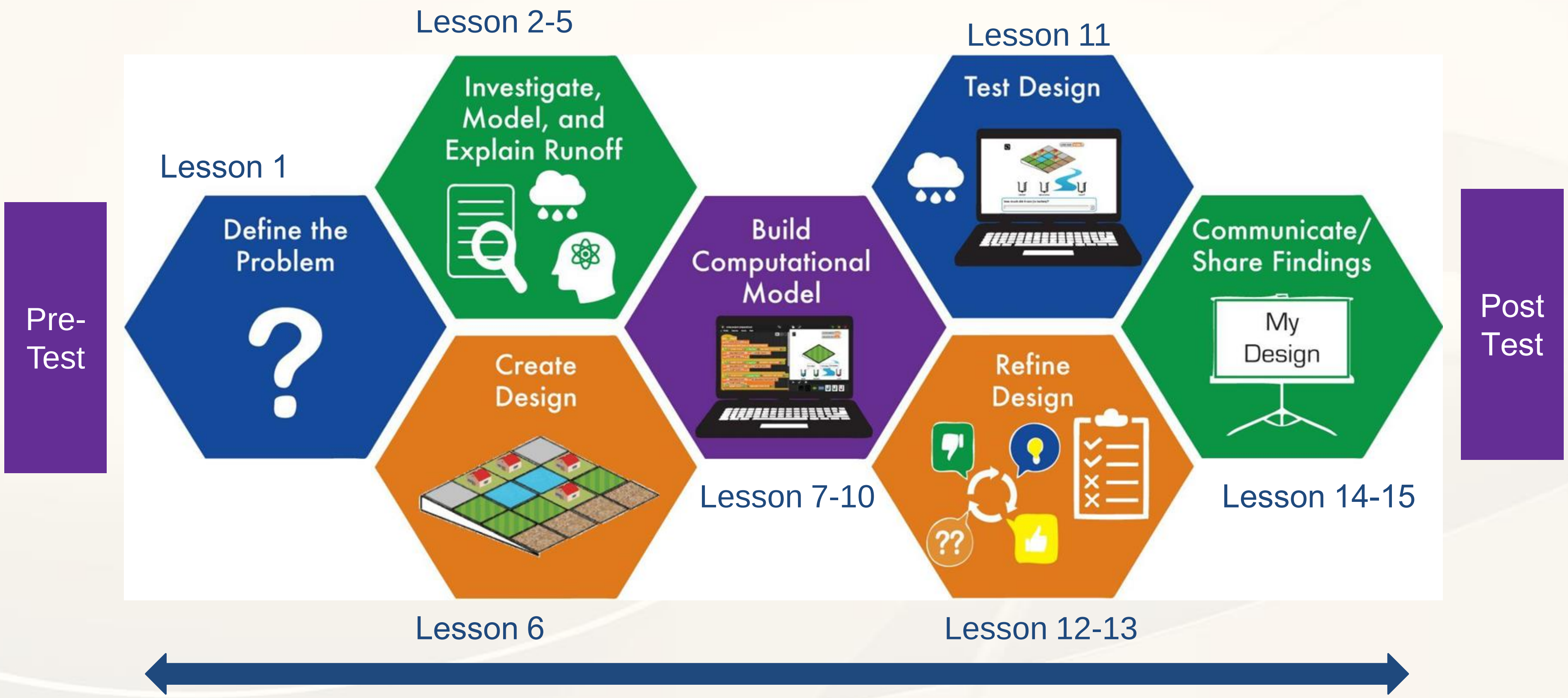


Testing Playground
Solutions

Input: rainfall
Output: runoff, cost

SPICE Curriculum

Integrated science and engineering anchored by computational thinking



- Conceptual modeling → Computational modeling → Engineering Design

Curriculum Development – Evidence-centered design (ECD) approach

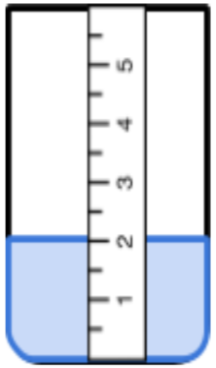
Investigate and Model

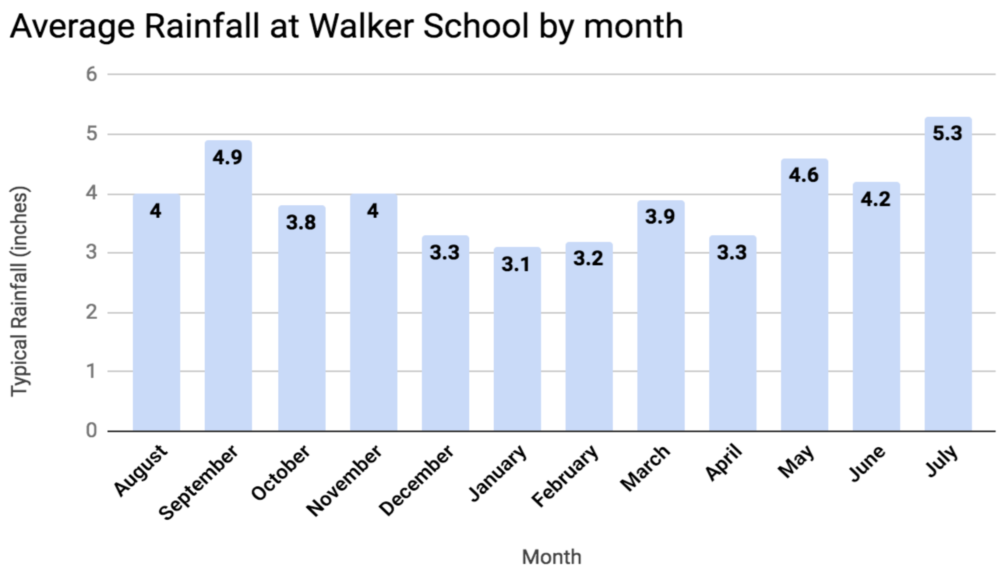
Lesson 2 : How much does it rain at Walker?

- Understand that rain is measured in inches
- Understand that heavy rainfall at Walker is more than 2 inches
- Understand the design criteria of 2 inches

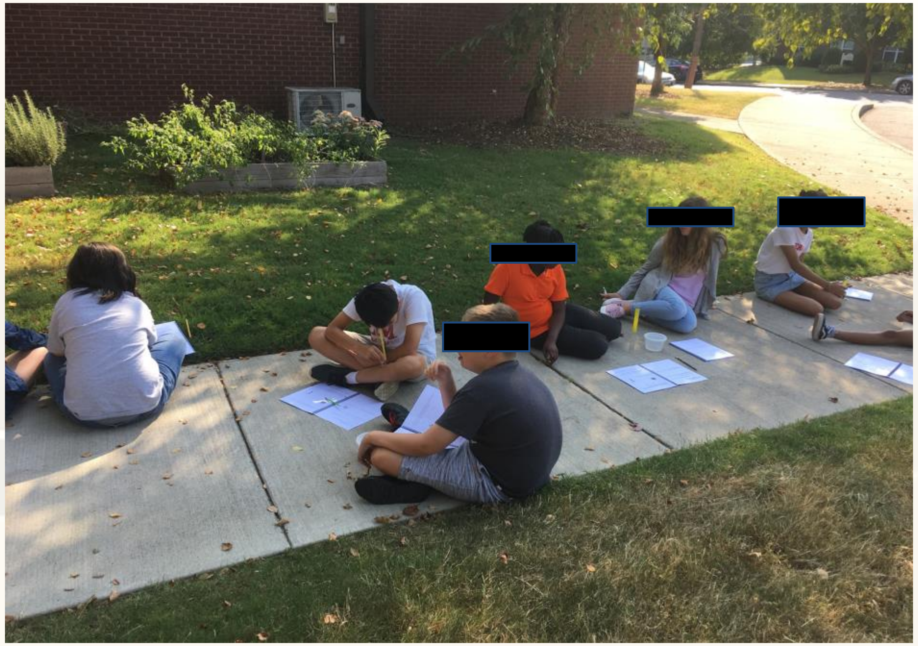
Discussion Question: How do we know how much water falls when it rains?

A rain gauge is used to measure the amount of rain that falls. Usually rain is measured in inches. For example, the rain gauge on the right shows **2 inches** of rain that fell in a heavy rainstorm:





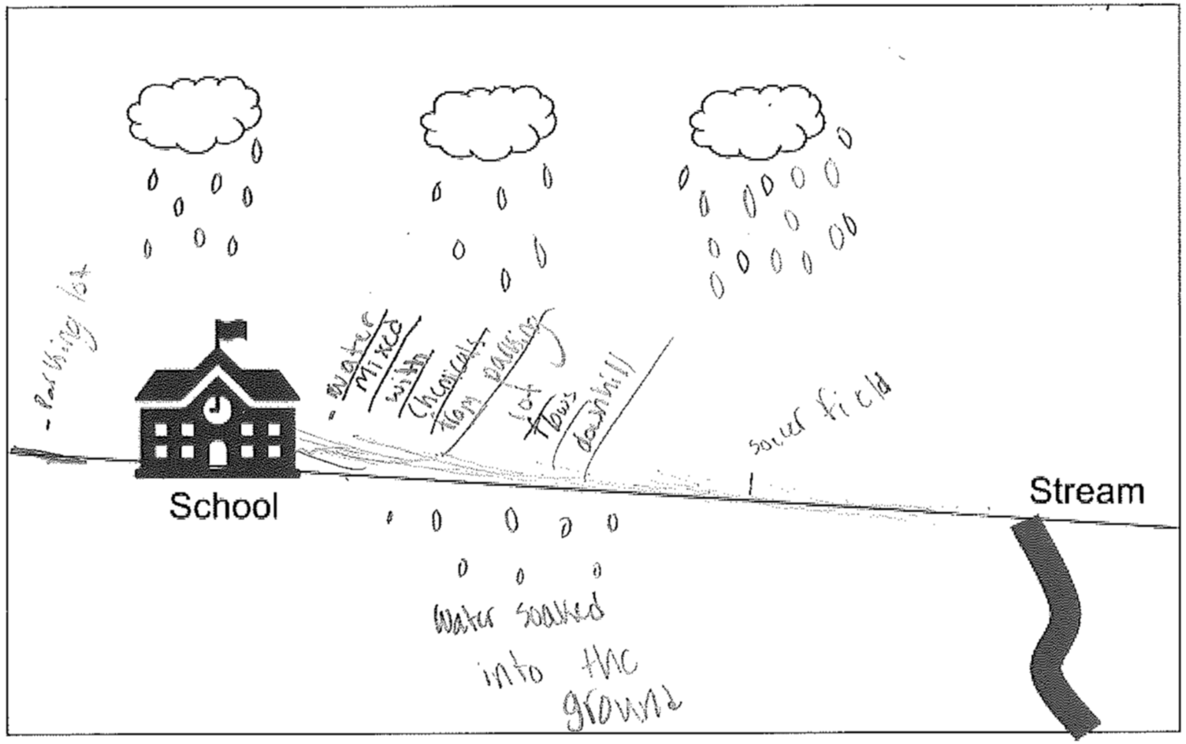
Lesson 3: Activity 1 – Hands-on Investigation



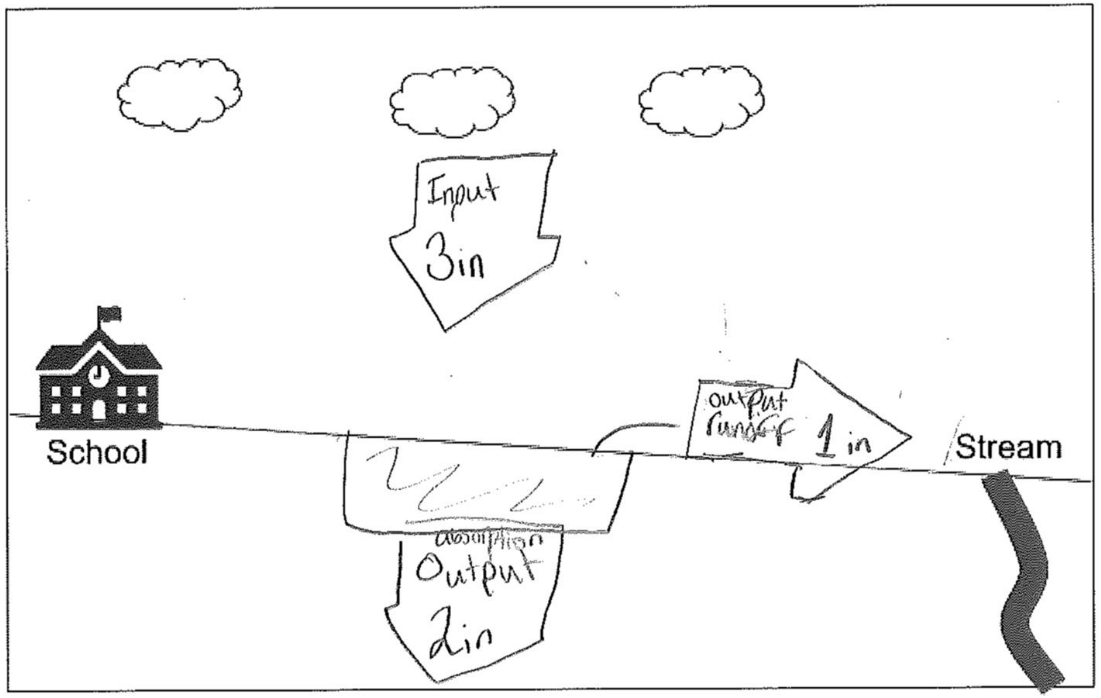
Lesson 4: Activity 2 – Conceptual Modeling

Pictorial representations

2.1. When it rains, where does the water go? Draw your best ***prediction*** below.
A **prediction** is a statement about what you *think* will happen based on observation.



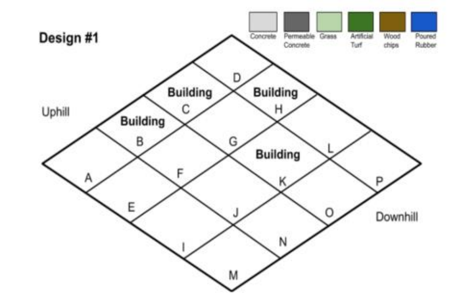
5.4. What if there are **3 inches of total rainfall**, but the ground's **absorption limit is 2 inches**? Draw a model showing the amount of total rainfall, total absorption, and total runoff.



From the physical to abstract modeling representations

Lesson 6: Create a Design

- Lesson 6 - How can we redesign Walker playground to reduce water runoff?
  - Become familiar with the available surface materials and their characteristics
  - Generate a design solution that meets some criteria
  - Recognize the need for a computer model to test their designs



Material	Description	Picture	Cost	Absorption limit	Accessible?
Standard Concrete	Poured material that hardens into a solid and seamless surface		$37,500 per square	Low	Yes (all students can use surface)
Permeable Concrete	Poured loosely packed material that looks and feels like concrete		$93,750 per square	High	Yes (all students can use surface)
Grass	Natural grass		$18,750 per square	High	No (not accessible to all students)
Artificial Turf	A carpet-like surface that looks and feels like grass		$112,500 per square	Medium	Yes (all students can use surface)
Wood Chips	Pieces of wood especially designed for playgrounds		$37,500 per square	High	No (not accessible to all students)
Poured Rubber	Rubber that can be poured into different shapes and colors.		$187,500 per square	High	Yes (all students can use surface)

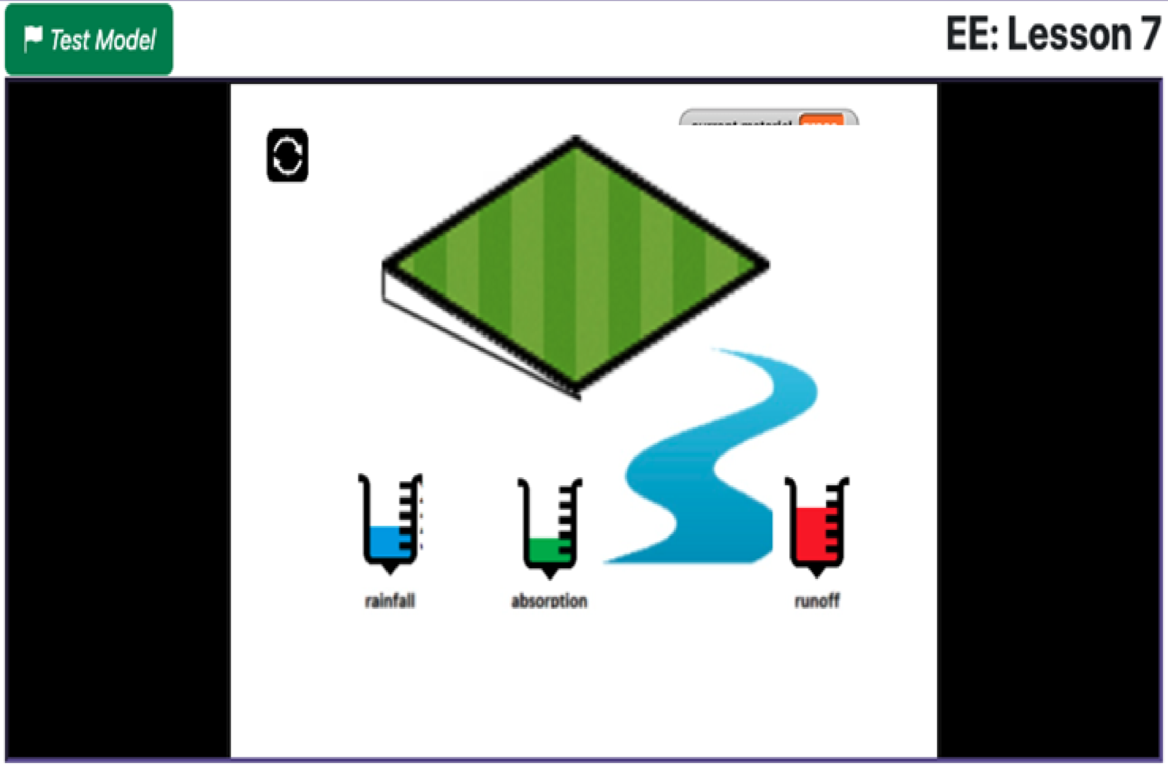
6.9. Make a design for Walker

Remember the criteria to minimize water runoff after heavy rains, stay under budget of $750,000, and the following:

Building	Grassy field	Play area	Parking	Accessible
4 squares (B, C, H, K)	At least 4 squares	At least 2 squares	At least 3 squares	At least 6 squares

Make Design #1:
- Label spaces with different purposes (grassy field, play area, parking).
- Color in what materials you choose using the key.
- Circle the accessible squares.

Lesson 7: Use Computational Modeling



Instructions | Test History

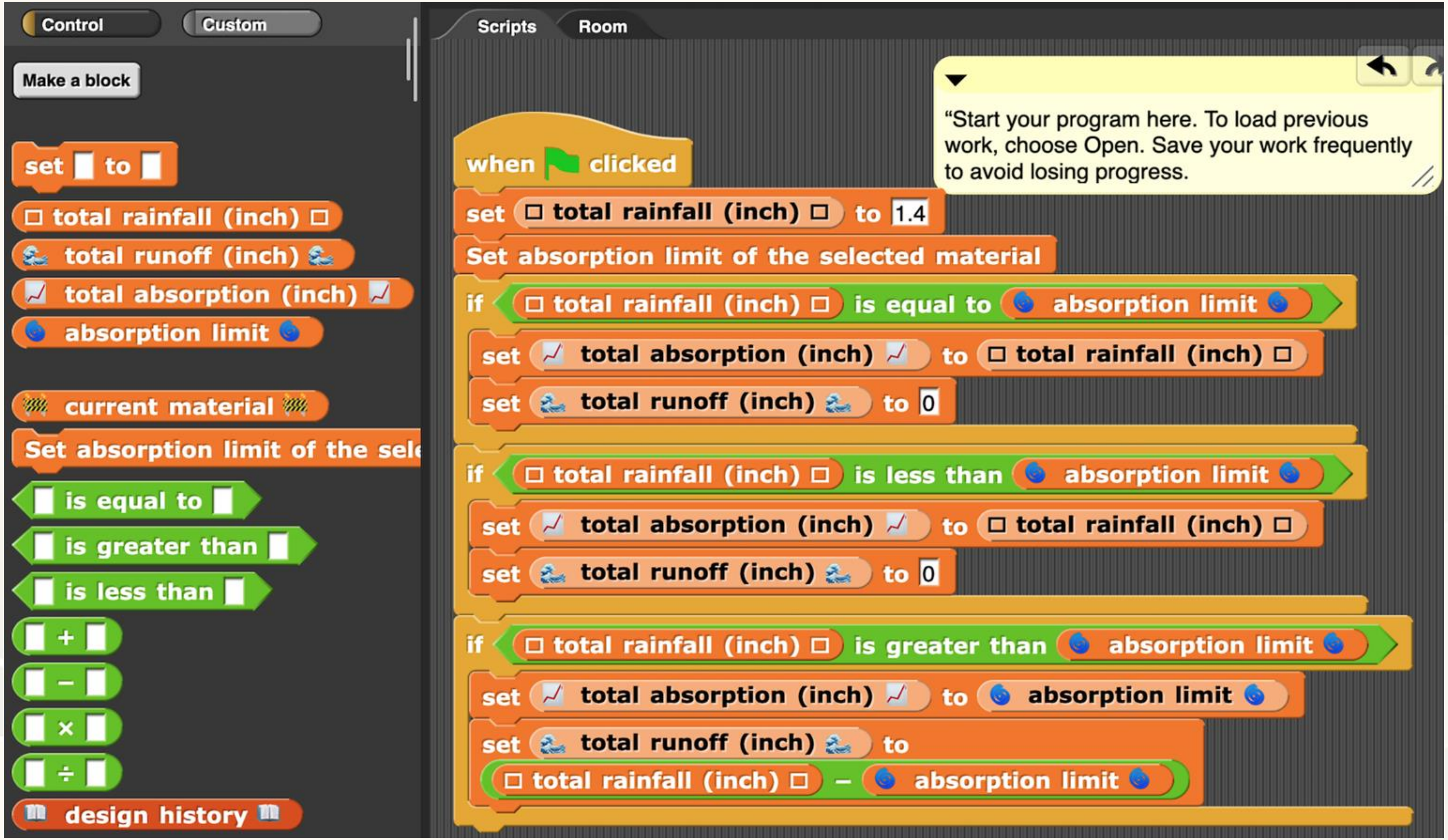
Instructions

- Click on the yellow tile to select a surface material.
- When you are ready to test the computational model, click on "Test Model."
- You will be asked "how much did it rain (in inches)?" In the textbox, insert a value for rainfall in inches (such as 1, 1.2, 2, or another number value).
- Press "Enter" on your keyboard or click the checkmark next to the textbox to test the model.
- Click on "Test History" to see all your results.

Read More..

Adopt a Use → Modify → Create approach

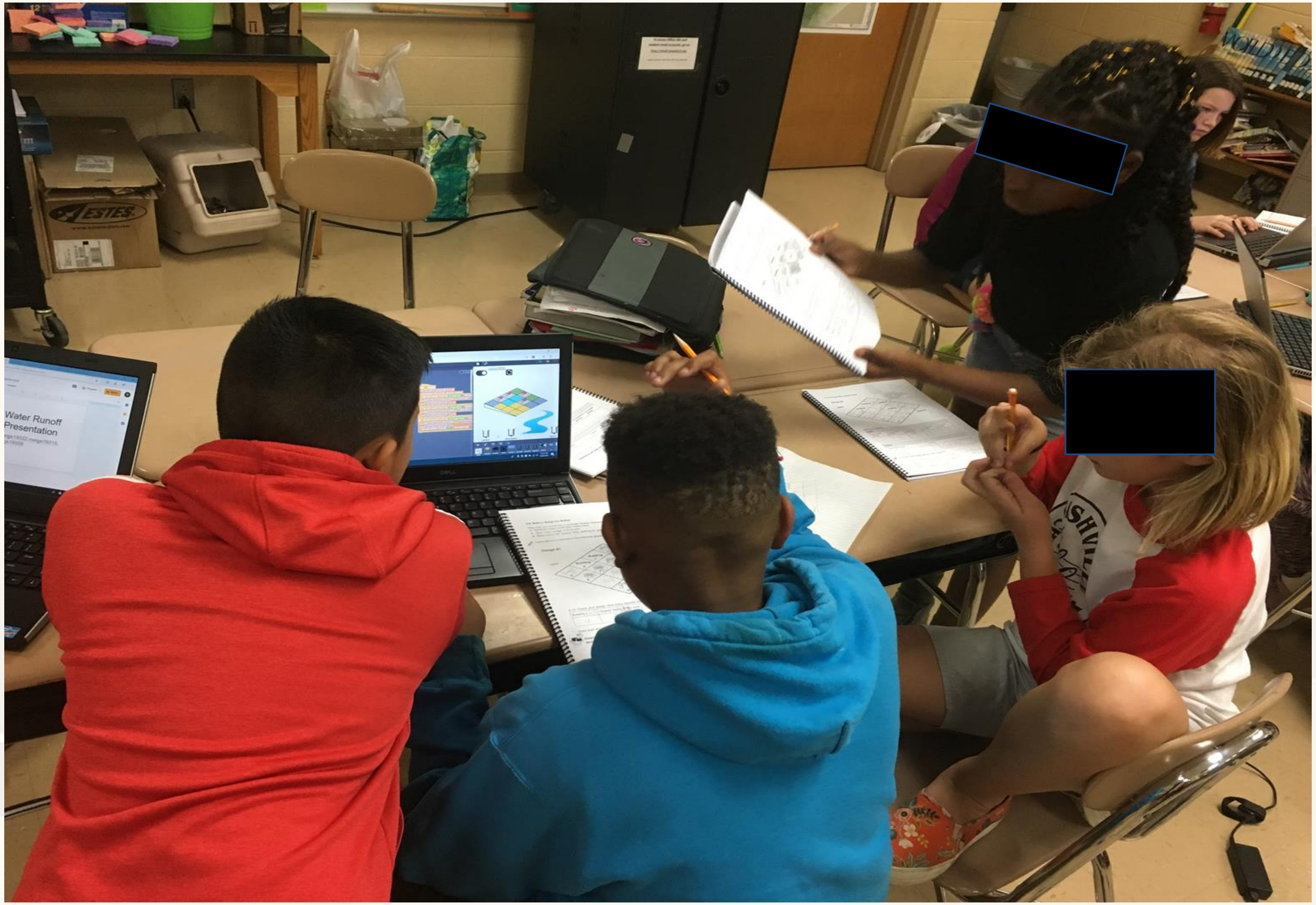
Lessons 8,9: Activity 3 – Computational Modeling



Integrating Science + CT

- Abstracted Domain-specific modeling language
- Students model three conditions that can occur after rainfall
- Students study how ground material affects absorption and runoff

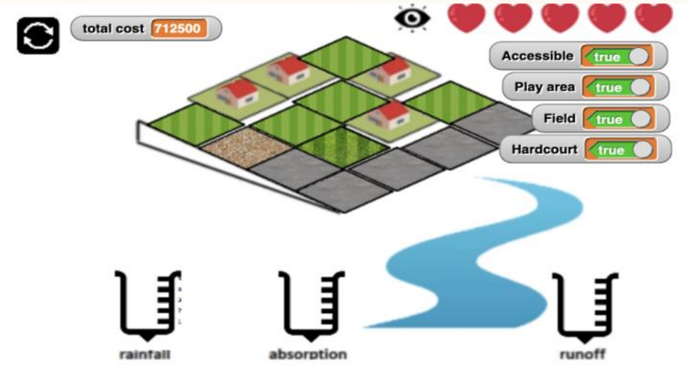
Activity 4: Solve the Design Challenge



Design Challenge
Minimize runoff and costs
Ensure accessibility

- Use computational model to find feasible design solutions
- Meet playground requirements and accessibility constraints
- Search for one that minimizes runoff and meets cost constraints
- Get together, discuss solutions, pick the best solution, and provide justification for the solution
- Present to the class

Lessons 11-13: Test & Refine Design



- Lesson 11 - How can we test and improve our designs?
  - Students test their designs and Use test results to evaluate design
  - Students generate multiple solutions.
- Lesson 12- How do you know what design will be the best?
  - Fair tests keep variables constant to evaluate multiple designs.
  - Conduct fair tests to compare designs based on specific criteria.
  - Designers often make trade-offs between variables, recognizing multiple perspectives on what constitutes the "best" design.
- Lesson 13 - How can you use the model to improve your design?
  - Designs can be improved through iterative testing and refinement
  - Documenting the results of design tests and comparing designs (using fair tests) can help to improve designs.

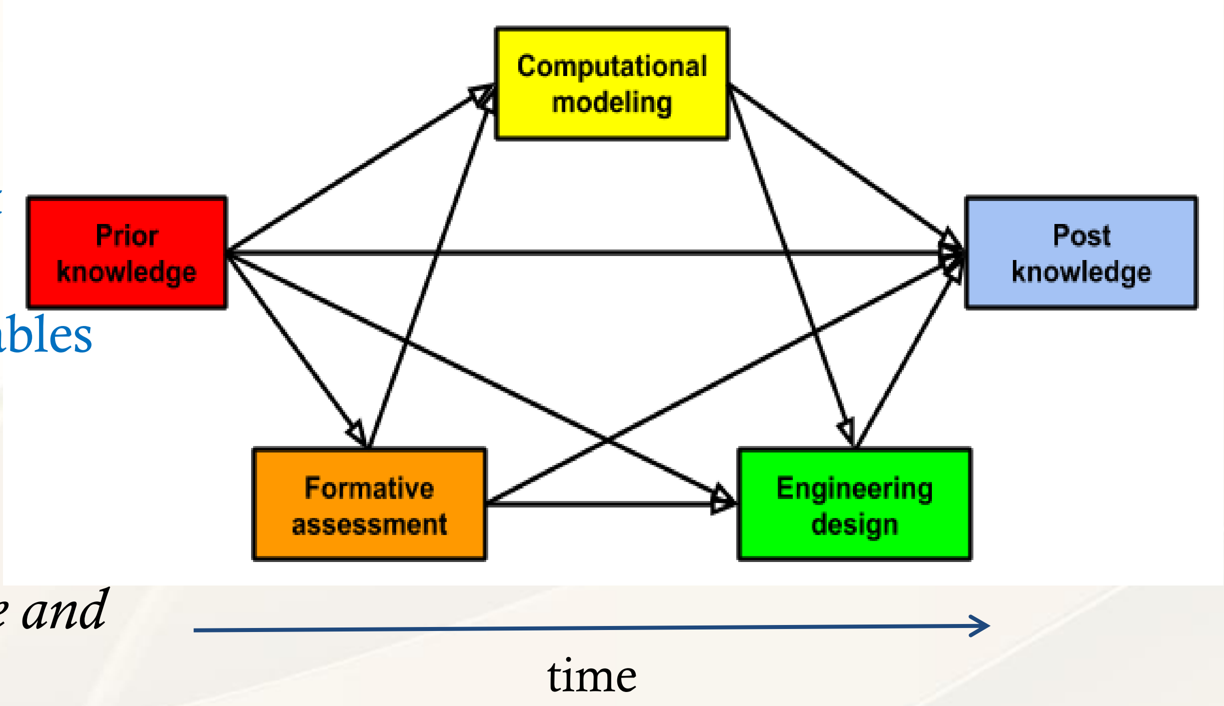
★	design/date	cost	runoff	accessible squares	rainfall	absorption	concrete	permeable concrete	grass	wood chips	artificial turf	poured rubber	compare
☆	5. 07/29/24 19:51:23	$693,750	1.07	6	2	0.9250	0	6	5	1	0	0	☐

Study

- Classroom study with 99 6th-grade students over 15 school days (14 lessons) in Fall 2019
  - The classroom study was led by two experienced science teachers
  - The two teachers received four days of professional development from the research team before the study
  - Three researchers provided additional support but mostly acted as observers
  - All participating students had varying amounts of prior programming experience with Scratch

Primary Analysis

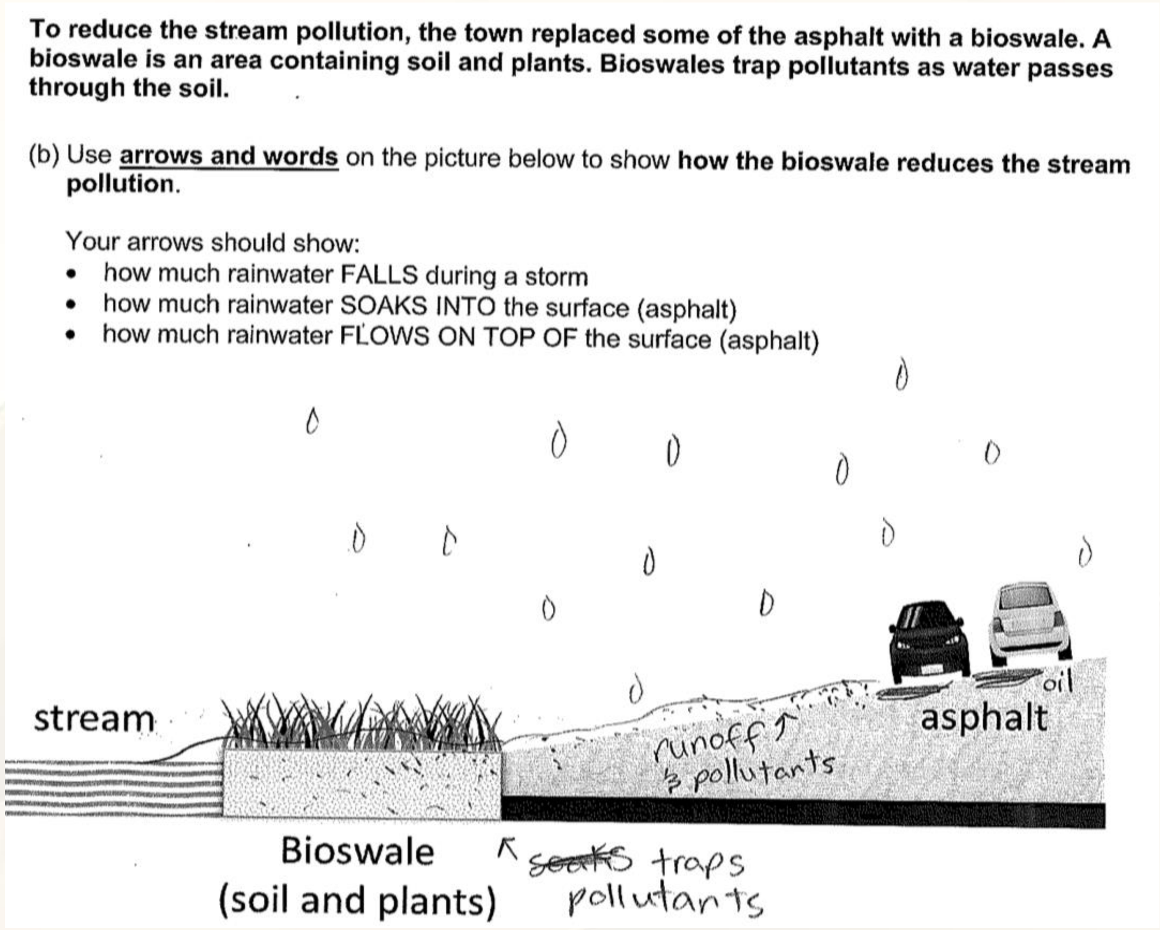
- Pre-post to study learning gains
- Path Analysis (Wright, 1983; Pearl & Mackenzie, 2018)
  - Similar to structured equation modeling (without latent variables) – multiple regression analysis
  - Study directed dependencies among a set of variables
  - Applied to study the effects among the measured performance and behavior variables



- What are the *relationships between learning science and performance in engineering design*?
- *What is the role of computational thinking in facilitating integrated science learning and engineering design?*

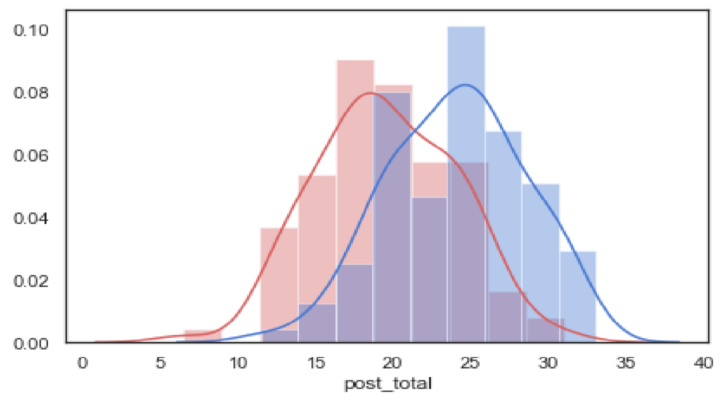
Methods

- Data collected from study
  - Science and engineering + CT pre-post assessments
  - Five formative assessments in science, engineering, and CT as homework
  - System logs of students' model-building activities
  - System logs of students' engineering design and testing activities using their own models
  - Exit Tickets



Pre-post Test Results

- All gains are statistically significant with a large effect size (*d* = 1.02)

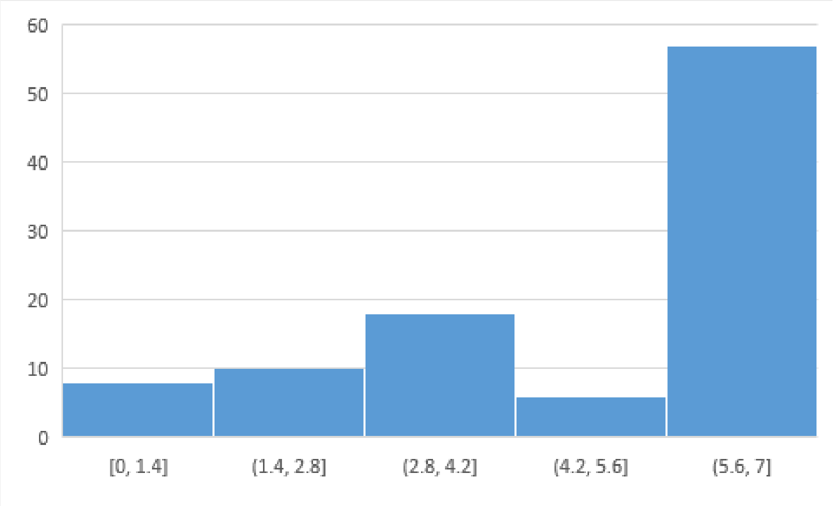


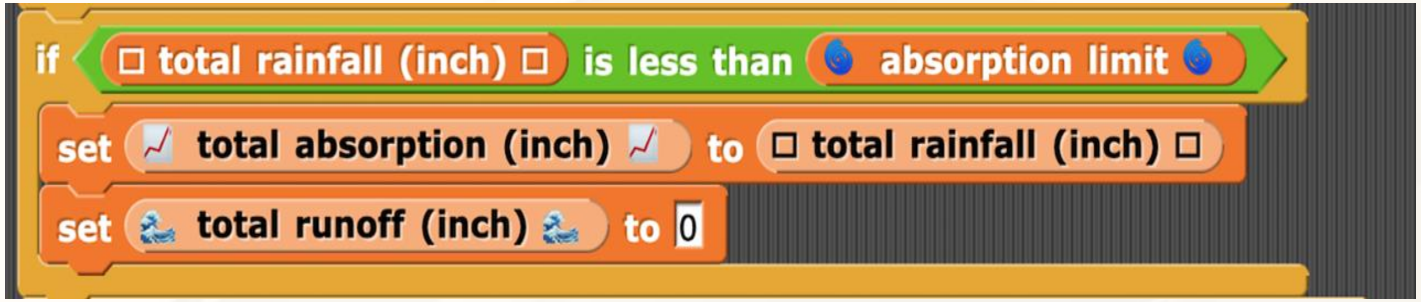
Plots (with kernel density estimation) of students' overall pre-post test score distributions – pre-test: *M* = 19.52 (*SD* = 4.47); post-test: *M* = 24.03(*SD* = 4.39)

	Total points	Pre-score(*stdev*)	Post-score(*stdev*)	*p*-value	Cohen's *d*
Science	7	4.56 (1.03)	5.13 (1.04)	<0.001	0.54
Engineering	16	8.73 (2.62)	10.50 (2.67)	<0.0001	0.67
CT	13	6.23 (2.60)	8.41 (2.69)	<0.0001	0.83
Overall	36	19.52 (4.47)	24.03 (4.39)	<0.0001	1.02

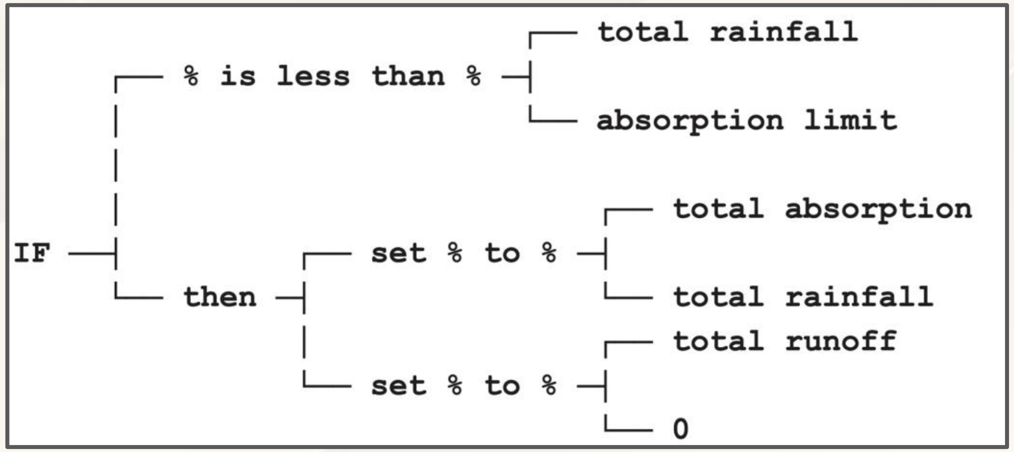
Computational Model scores

- The average computational model score was **4.67 (SD = 1.85)** [max score = 6]
- **59%** of the students created a correct computational model before the correct model was discussed in class



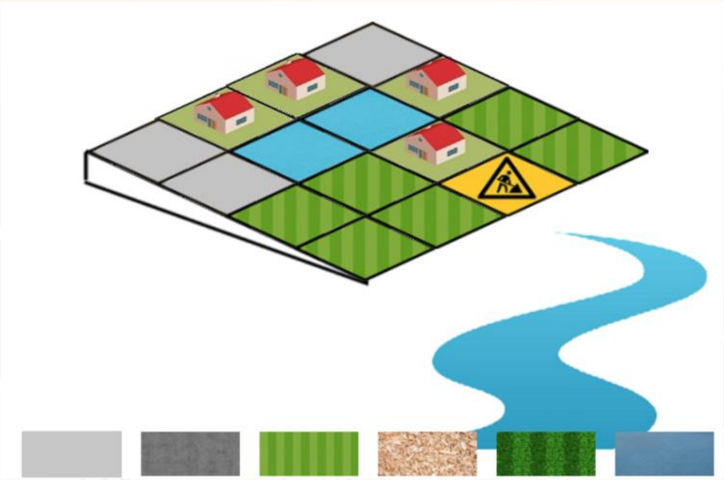


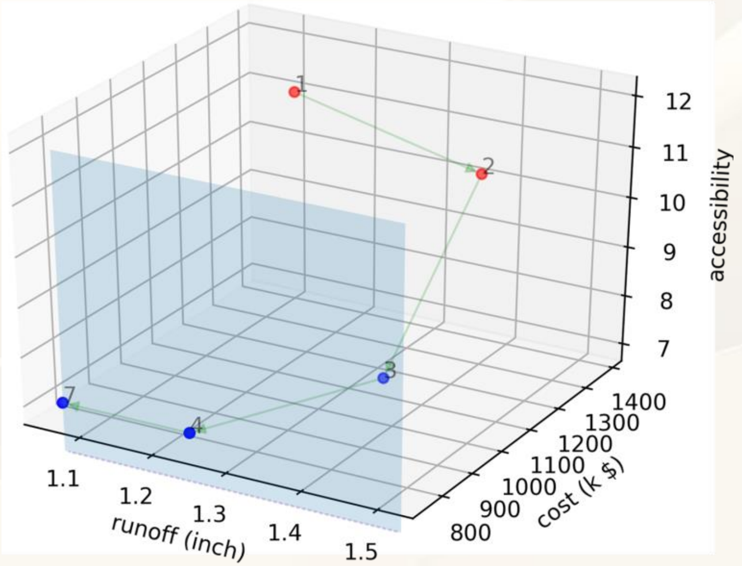
	computational model edits	computational model tests	computational model score
mean	432.4	31.9	4.67
std	211.4	20.5	1.85

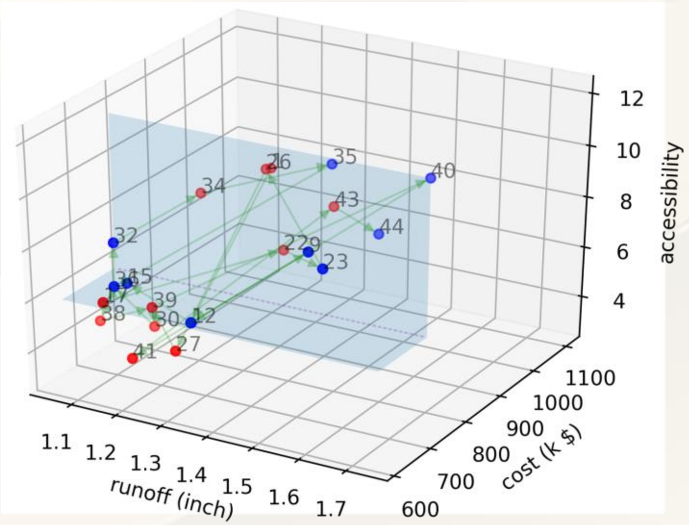


Evaluating Design Solutions

- Measurements of students' learning activities during the computational model-building and engineering design activities
  - The number of tested designs, the number of satisfying designs, and the (normalized) Euclidean distance between tests



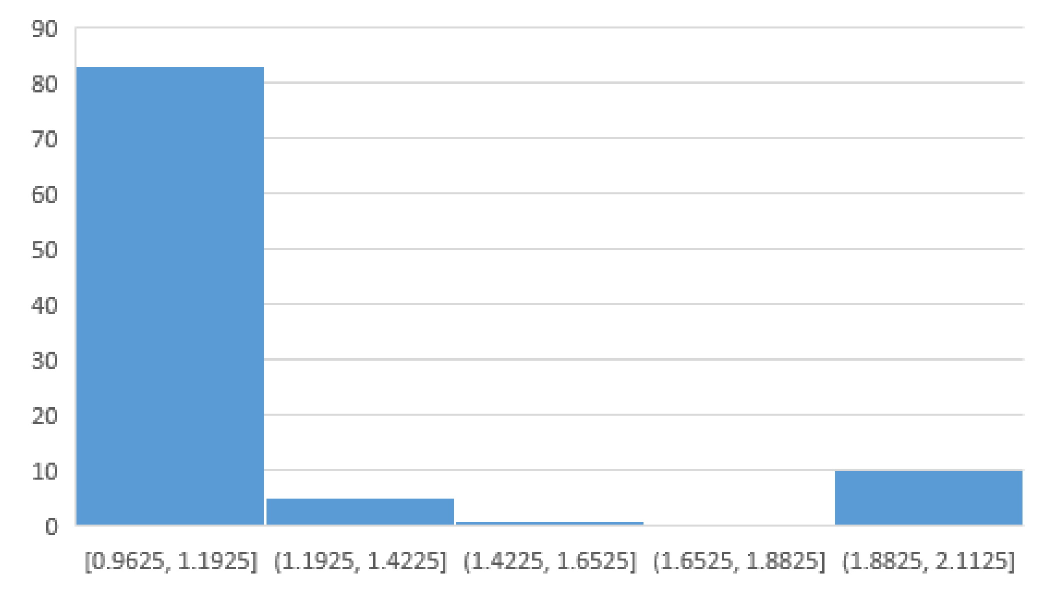




Evaluation of Design solutions

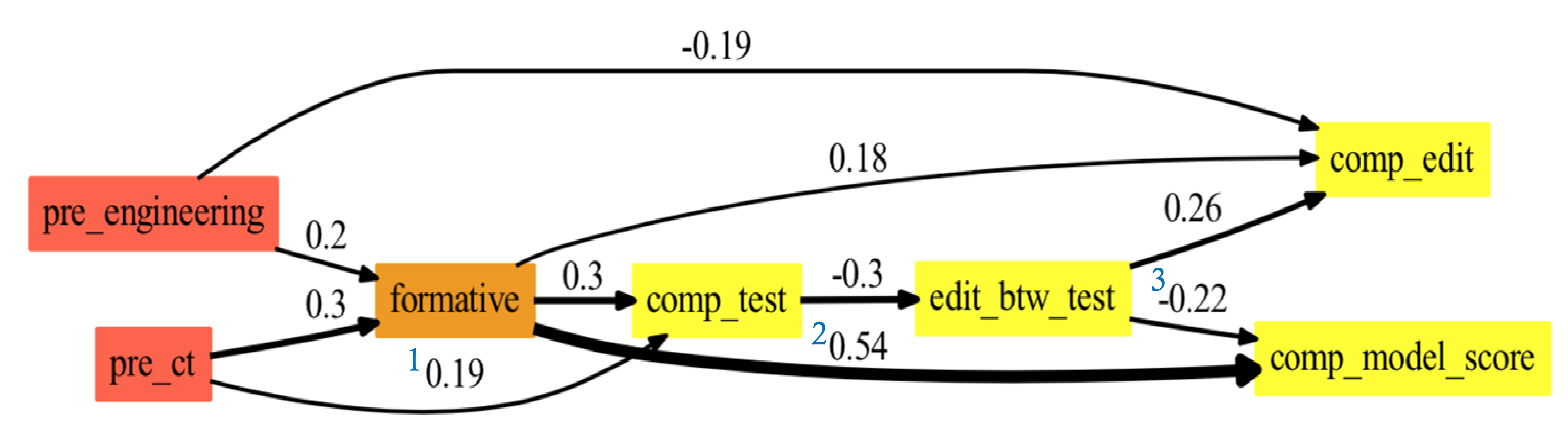
- The average number of unique designs that satisfied the criteria for cost and accessibility was **6.3 (SD = 4.2)**
- **89 students** created and tested at least 1 satisfying design
- the global minimal runoff of all satisfying designs was 0.9625 inches, and **29 students** arrived at this optimal solution

	engineering test	unique satisfying design	lowest runoff
mean	29.38	6.31	1.23
std	22.19	4.25	0.94



Path Analysis Results

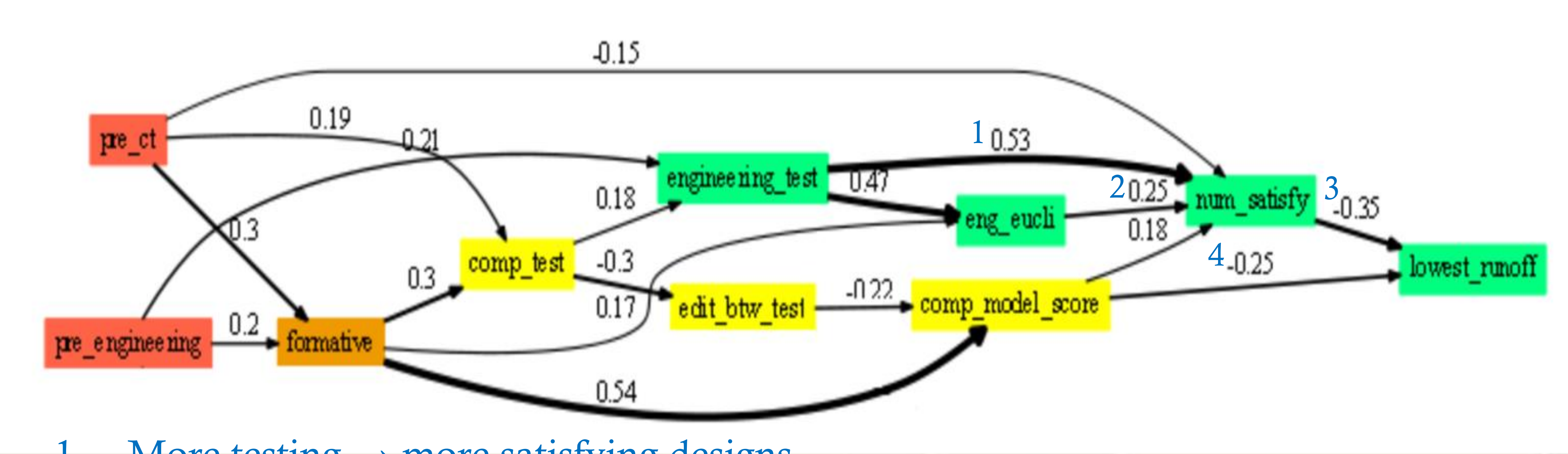
- What affected computational model building performance?



1. Higher CT knowledge → more testing
2. Higher integrated proficiency (formative assessments) → higher model score
3. Smaller edit chunks (edit_btw_tests) → higher model score

Path Analysis Results (2)

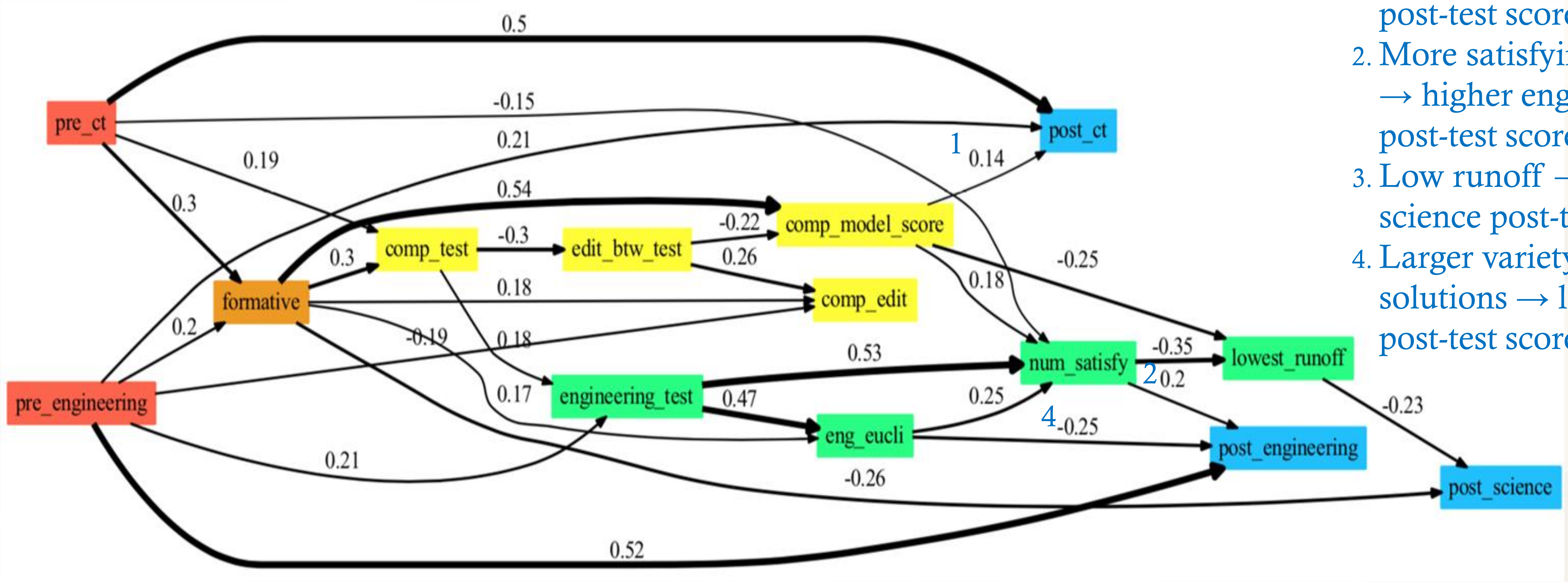
- What affected the engineering design measures?



1. More testing → more satisfying designs
2. Larger total Euclidean distance → more satisfying designs
3. More satisfying designs → lower runoff (better performance)
4. Higher computational model score → lower runoff (better performance)

Path Analysis Results (3)

- Effects on the post-test scores (overall learning)



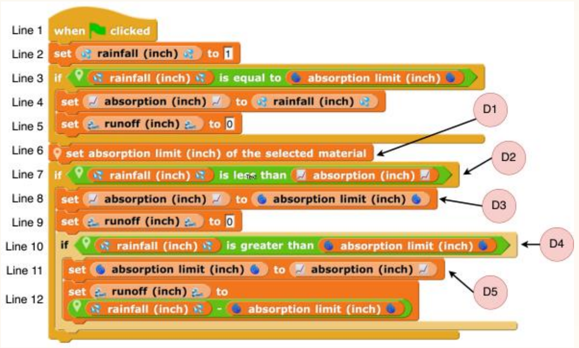
1. Higher computational model score → higher CT post-test score
2. More satisfying designs → higher engineering post-test score
3. Low runoff → higher science post-test score
4. Larger variety of engg solutions → lower engg post-test score

CONVERSATIONAL AGENT – INQUIZZITOR

Automated Assessments and Feedback to Support Teachers and Students

https://faagent.c2-stem.org

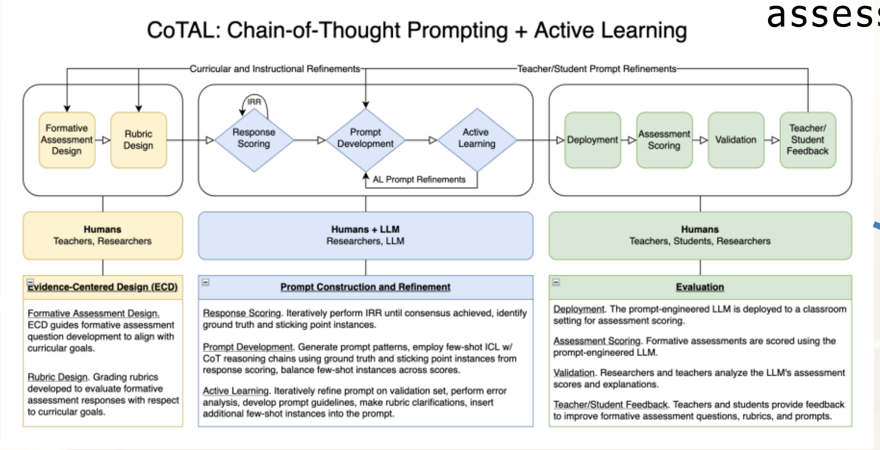
Supporting Teachers & Students: Inquizzitor



Students take a formative assessment

Students' agent interactions provide information to teachers and refine our assessments, rubrics, and prompts

Students interact with Inquizzator, gaining a more comprehensive understanding of their scores, & actionable guidance to improve their conceptual understanding and correct their misconceptions.



Human-in-the-loop prompt engineering with GPT-4o, in-context learning, chain-of-thought reasoning, and active learning (CoTAL)

Using principles from socio-cognitive theory: encouraging self-regulation and self-efficacy, we developed a formative assessment agent powered by GPT-4o and Gradio

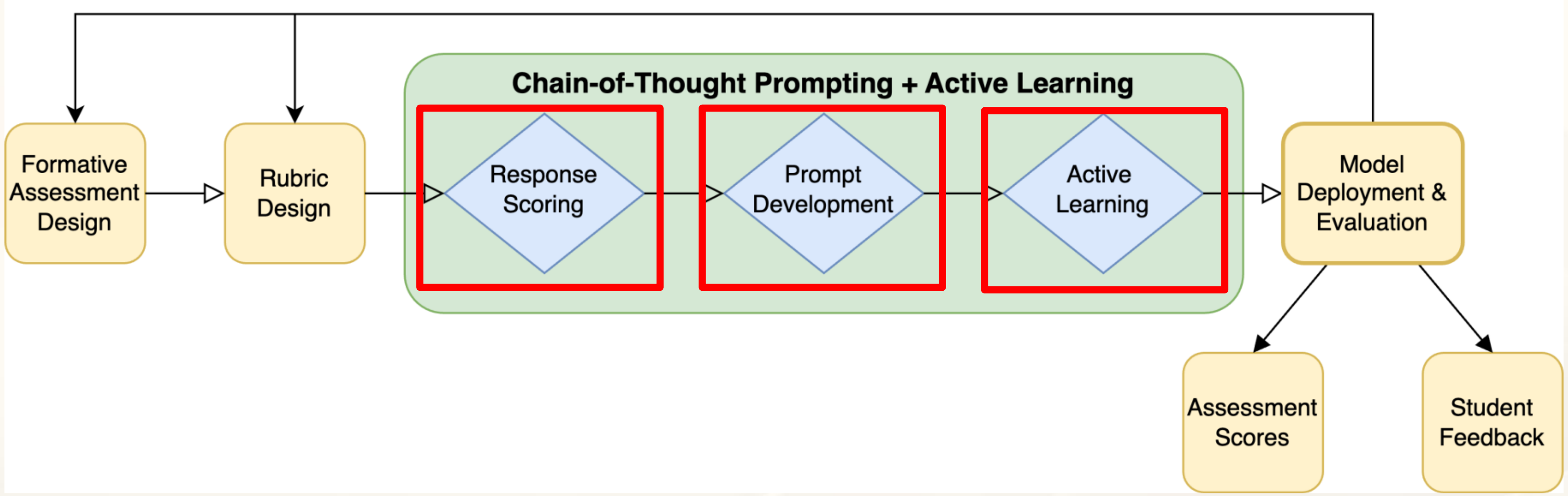
Supporting Teachers & Students

Automated Grading of Formative Assessments

- In-Context Learning (ICL).
  - An emergent behavior in LLMs where the model learns a new task at inference time (i.e., without parameter updates) via labeled few-shot examples in the prompt (Brown et al., 2020).
- Chain-of-Thought Reasoning (CoT).
  - An extension of ICL that uses a series of intermediate reasoning steps in the few-shot examples to guide the LLM toward the correct solution, improving LLM performance over traditional ICL (Wei et al., 2022).
- Active Learning (AL).
  - A process by which a learning algorithm can interactively query a human-in-the-loop (an "oracle") to label new instances (particularly hard-to-predict ones) for improved training.

Method

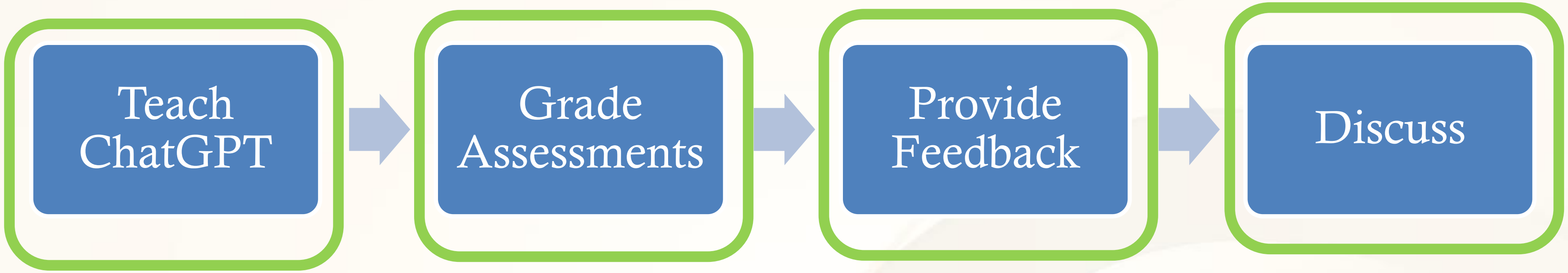
- CoTAL – Chain of Thought and Active Learning



- human-in-the-loop approach consists of three steps and combines chain-of-thought prompting with active learning to
  1. Align the LLM with the humans' scoring consensus, and
  2. Provide meaningful feedback by explaining the scoring

Supporting Teachers & Students

Automated Grading of Formative Assessments



- Researchers teach ChatGPT to grade student responses like a teacher would, showing examples that explain the thinking behind the rubric
- Once humans validate ChatGPT's responses for accuracy, ChatGPT uses the examples to guide its scoring, making sure each grade matches the reasoning in the rubric
- Students receive their scores via our formative assessment agent, **Inquizzator,** along with clear explanations to help them spot gaps in their understanding and learn from mistakes
- Once humans validate ChatGPT's responses for accuracy, ChatGPT uses the examples to guide its scoring, making sure each grade matches the reasoning in the rubric
- Students engage in a follow-up discussion with the agent to deepen their understanding and get personalized guidance on what to learn next

Example Formative Assessment: Fair Testing

- Morgan has two designs and wants to know which design is better.

Morgan tested her **FIRST** design with these **inputs and outputs**:

Inputs					
Rainfall (inches)	**# Building squares**	**# "Grassy" squares**	**# Play squares**	**# Parking squares**	**# Accessible squares**
6	6	4	3	3	8

Outputs	
Runoff (inches)	**Cost ($)**
4.8	$732,000

Morgan tested her **SECOND** design with these **inputs and outputs**:

Inputs					
Rainfall (inches)	**# Building squares**	**# "Grassy" squares**	**# Play squares**	**# Parking squares**	**# Accessible squares**
1	4	3	4	5	5

Outputs	
Runoff (inches)	**Cost ($)**
0.0	$695,000

Fair Testing

1. Based on the above tests, can these designs be compared fairly?*

- ☐ Yes
- ☐ No

2. Explain why or why not:*

Your answer ______________________________

Trade-Offs

3. Describe a trade-off regarding something that happened or is currently happening in your own life or that you've learned about in school?

For instance, you could compare virtual learning and in-person learning, or picking between extracurricular activities.

Your answer ____

Fair Testing Rubric

- ***Based on the above tests, can these designs be compared fairly? Explain why or why not.***
  - 0 points: responded “Yes” that the designs can be fairly compared.
  - 1 point: responded “No” that the designs can be fairly compared but did not provide an explanation related to fair tests or design constraints.
  - 2 points: responded “No” that the designs can be fairly compared, AND discussed design constraints such as runoff, accessible squares or cost.
  - 3 points: responded “No” that the designs can be fairly compared AND discussed the different rainfall amounts as the reason for this.
  - 4 points: responded “No” that the designs can be fairly compared, discussed the different rainfall amounts as the reason, AND mentioned that having unequal rainfall amounts results in the tests not being **fair**.

Case Study

Student did not receive a perfect score

AGENT: Hi there!

My name is Inquizzitor, and I'm here to help you explore your assessment responses. Think of me as your guide to better understanding the material and clearing up any tricky spots!

Below, you'll find my feedback on your responses. Feel free to ask me about the scores I gave you or chat with me about anything you're unsure of. Let's work together to sharpen your skills and tackle any challenges!

Student Assessment Response:

Answer: No

Explanation: She used more rain for design 1 so the one that is the best is not clear.

Inquizzator Feedback: ***ChatGPT praises the student for the portion of the question correctly answered.***

Great job recognizing that the two designs cannot be compared because 'She used more rain for design 1'. This shows you understand the importance of consistent testing conditions. To improve your response, consider explaining why using different amounts of rain makes the tests unfair. Keep up the good work!

- **Score**: 3/4

ChatGPT explains why the student did not receive full points.

Case Study (contd. …)

Student did not receive a perfect score

STUDENT: how could i improve *The student asks ChatGPT how to improve his or her response.*

AGENT: Great question! Here's how you can improve your response to achieve full credit:

- **Mention Fairness**: Explain that using different amounts of rainfall makes the tests unfair. You can say something like, "The tests are not fair because the different rainfall amounts make it impossible to compare the runoff results accurately."

- **Engineering Constraints**: You could also mention that each design violates one of the Engineering Constraints. For example, "Design1 does not minimize runoff, and Design2 does not have at least 6 accessible squares."

ChatGPT provides specific suggestions, along with relevant examples.

By adding these points, your response will be more complete and demonstrate a deeper understanding of why the designs cannot be fairly compared.

Keep up the good work, and you'll get there!

ChatGPT encourages the student to keep at it.

Benefits

- Unlike traditional automated assessment scoring using black-box AI algorithms, Inquizzator:
  - Explains *why* students received their scores, based on the rubric and examples provided by researchers and teachers
  - Provides recommendations and examples for improvement
  - Allows for dynamic, multi-turn conversations for students to deepen their conceptual understanding
  - Provides actionable next steps
  - Allows for increased user engagement

Potential Drawbacks

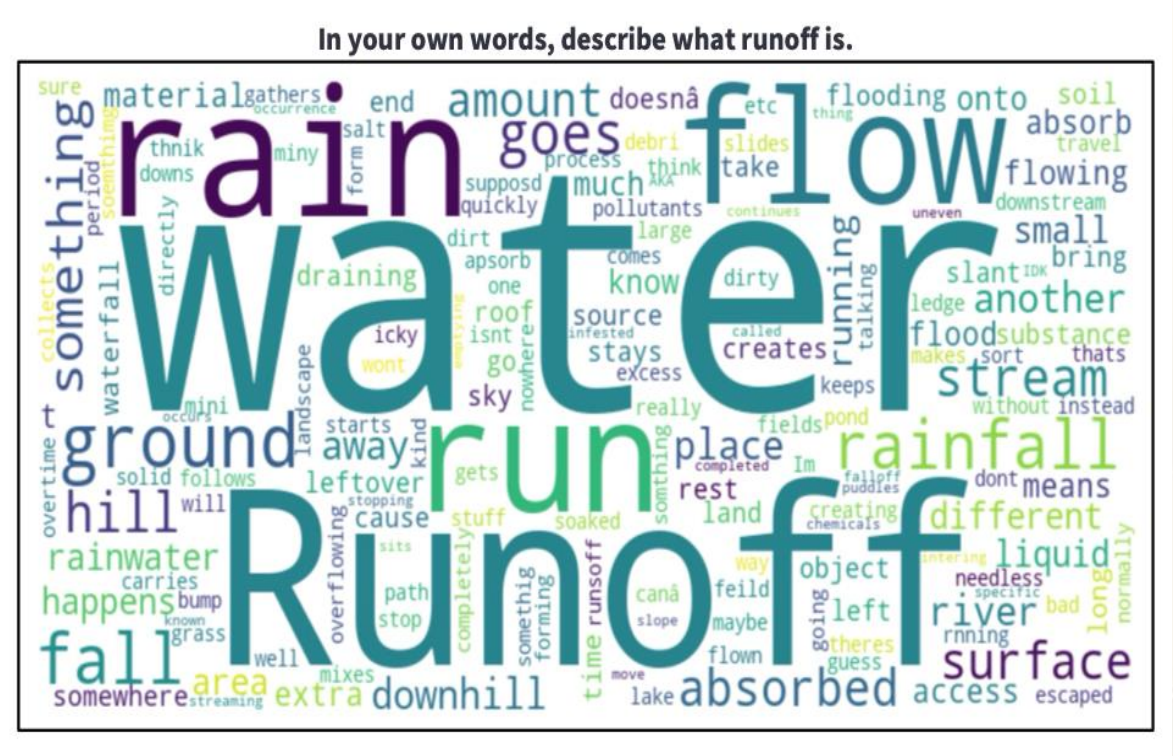
- Bad actors. Students, generally, have experience with ChatGPT and may attempt to subvert its guardrails
- Off-topic discussion. Many students in our study focused on how to “break” Inquizzator rather than deepening their understanding of the course material
- Score fixation. Students often focused on trying to convince Inquizzator to increase their scores rather than understanding why their assessment responses were inadequate

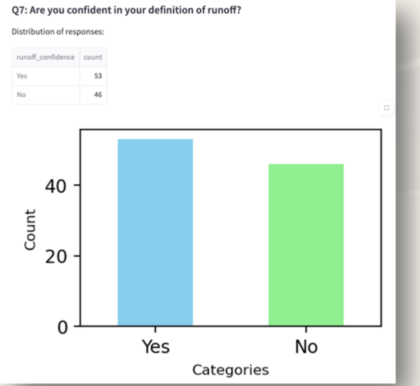
TEACHER DASHBOARD

ANALYZING EXIT TICKETS USING AI

Supporting Teachers: Dashboards

- Features
  - Question by Question Analysis of Exit Tickets
    - Sample responses
    - Word cloud of responses
    - Bar-chart
    - AI-generated summary of responses

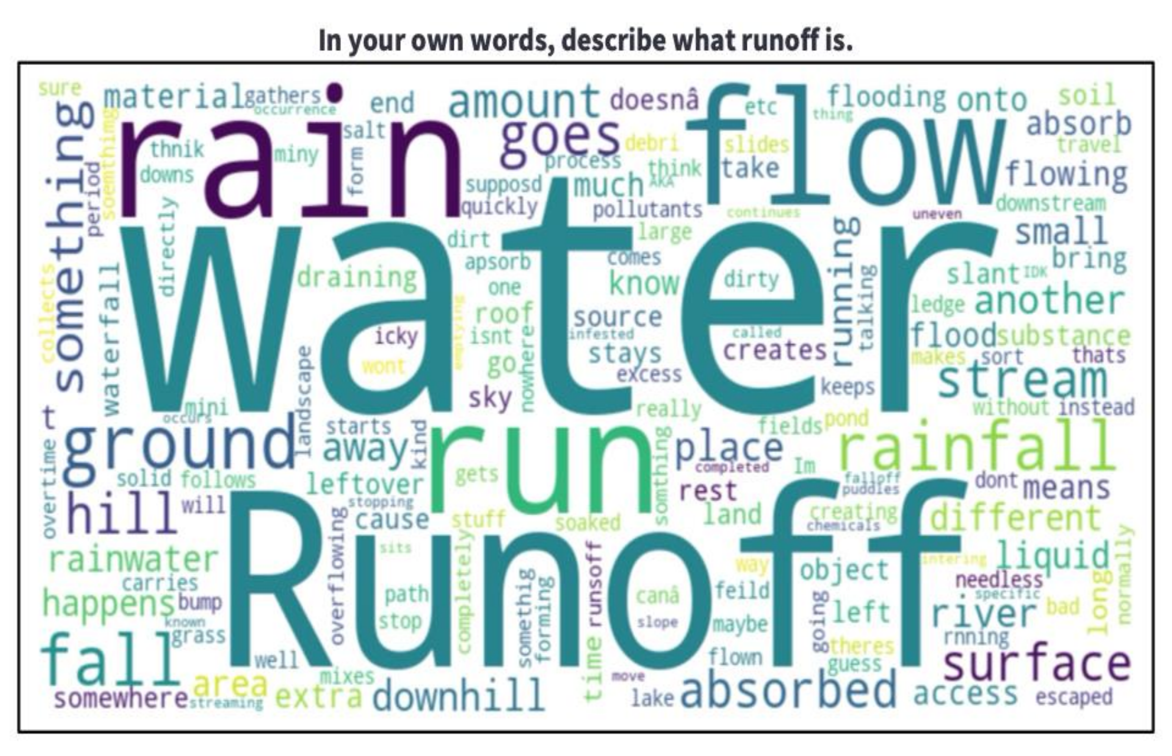




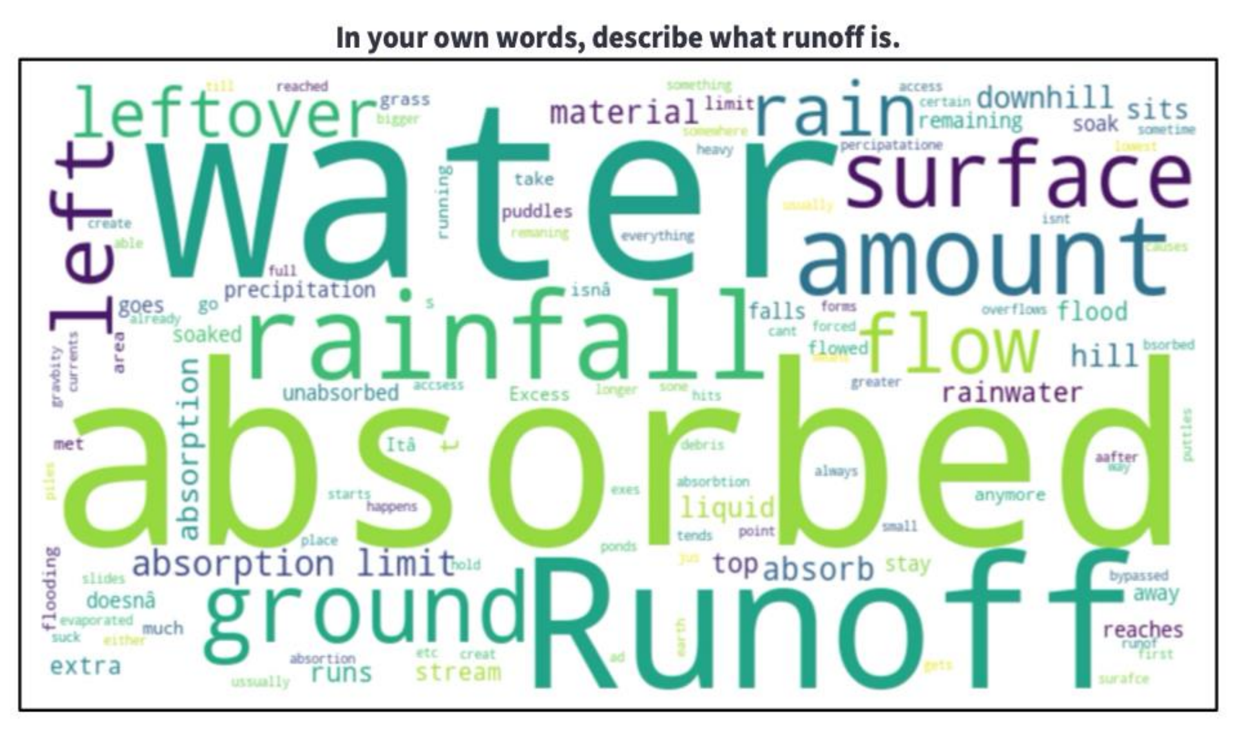
runoff_confidence	count
Yes	53
No	46

Supporting Teachers

Student understanding of runoff



Lesson-1



Lesson-15

Word Cloud Representation

Conclusions & Future Work

- Demonstrated the interconnectedness between science, engineering, and CT in an NGSS-related curriculum (Zhang, et al, 2019; 2020); Hutchins, et al,)
- Automated Analysis of Assessments
  - Conversational Agent powered by LLMs to support teachers and student learning (Cohn, et al, 2024; in submission)
- Initial Prototypes of Teacher Dashboards
  - Visualization of Analytics (Hutchins & Biswas, 2024); Feedback on Formative Assessments (Cohn, et al, 2023; 2025); Exit Tickets (Srivastava, et al, 2025)
- Future work
  - Better measures for evaluating students' model building and engineering design
  - Further development of adaptive scaffolding using conversational agents to support student learning

Acknowlegments

Shruti Jain
Ph.D. student

Clayton Cohn
Ph.D. student

Namrata Srivastava
Research Scientist

Naveed Mohammed
Sr Research Engineer