Large Language Models

Transformational Technology

The world will operate much differently in ways we may not be able to predict

Enabling Technology

Make humans more efficient

Create starting drafts of text or code

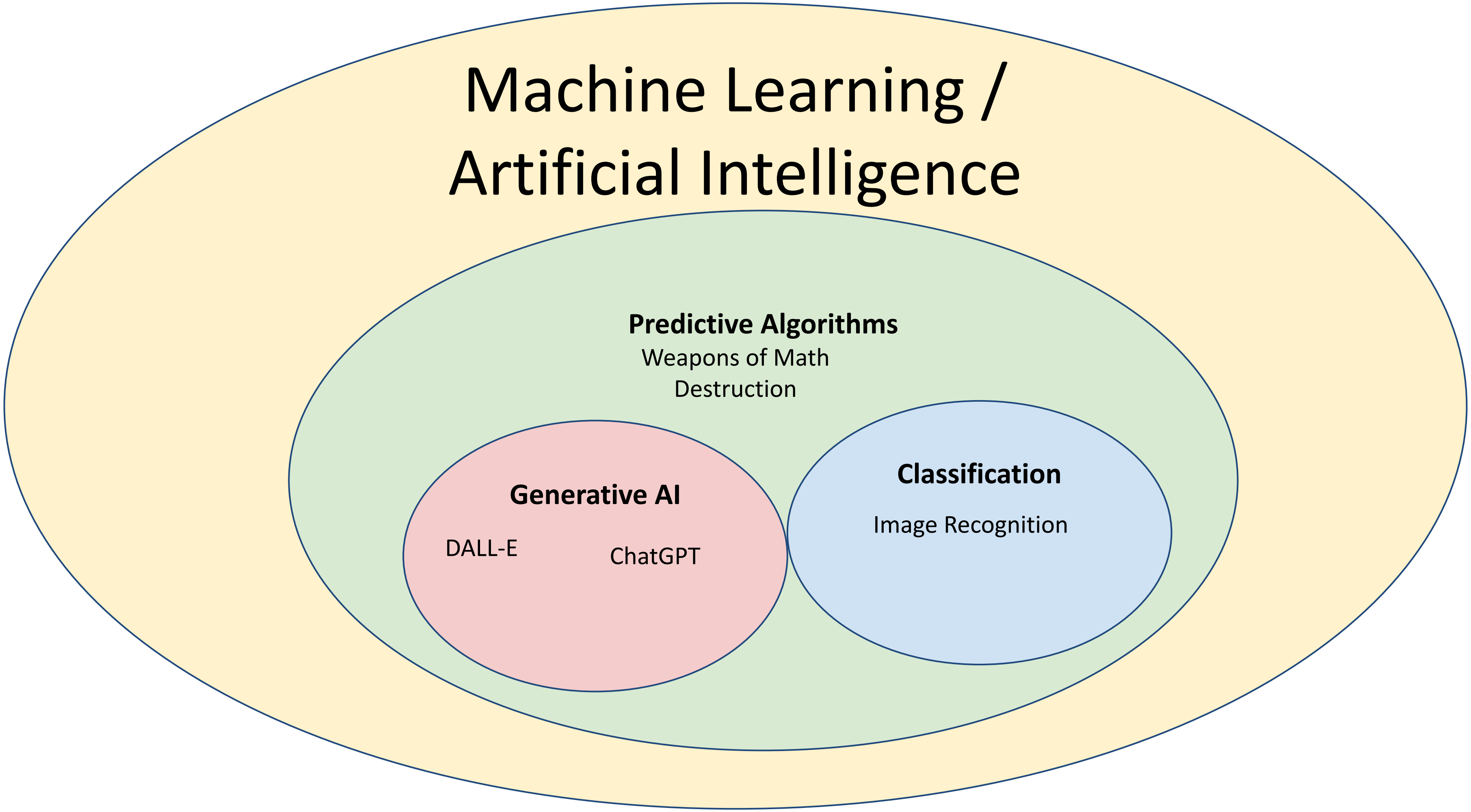
Disruptive Technology

Replaces jobs

More successful at customer service in one service center

Simple artwork

Terms / Context



Machine Learning / Artificial Intelligence

Predictive Algorithms

Weapons of Math Destruction

Generative AI

DALL-E

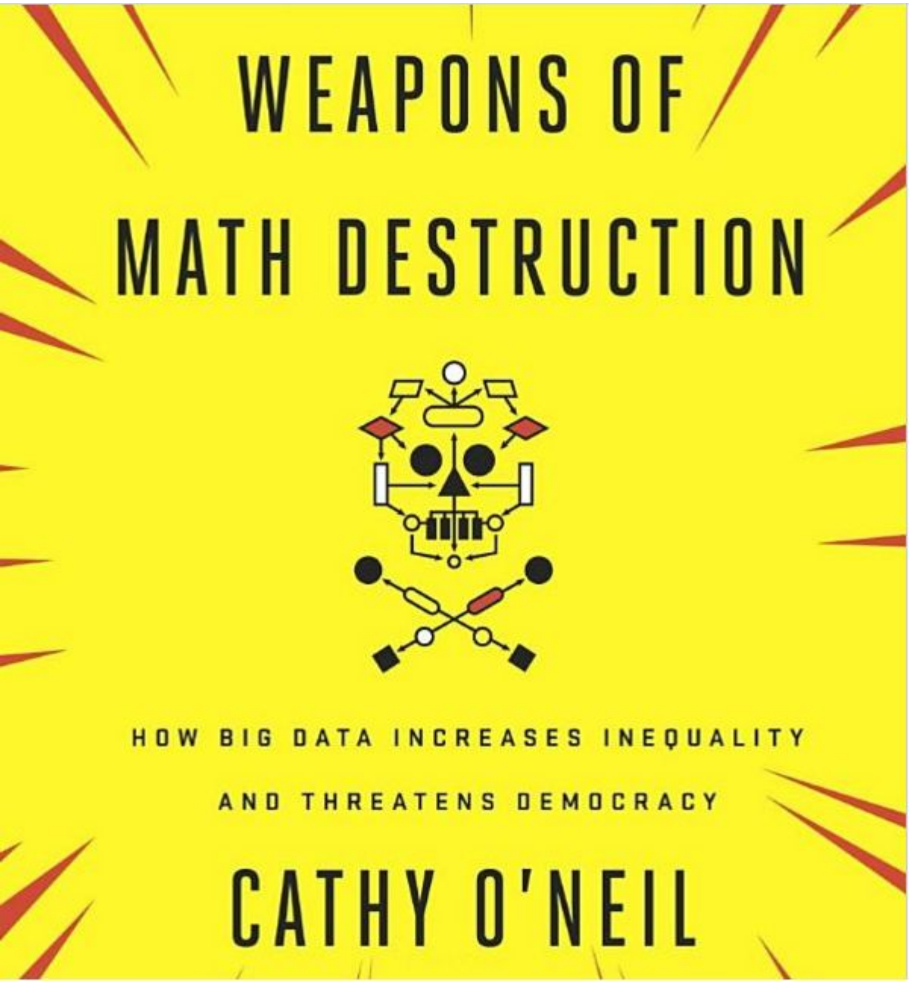
ChatGPT

Classification

Image Recognition

Weapons of Math Destruction

How Big Data Increases Inequality and Threatens Democracy



Cathy O’Neil
(c/o Diana Franklin)

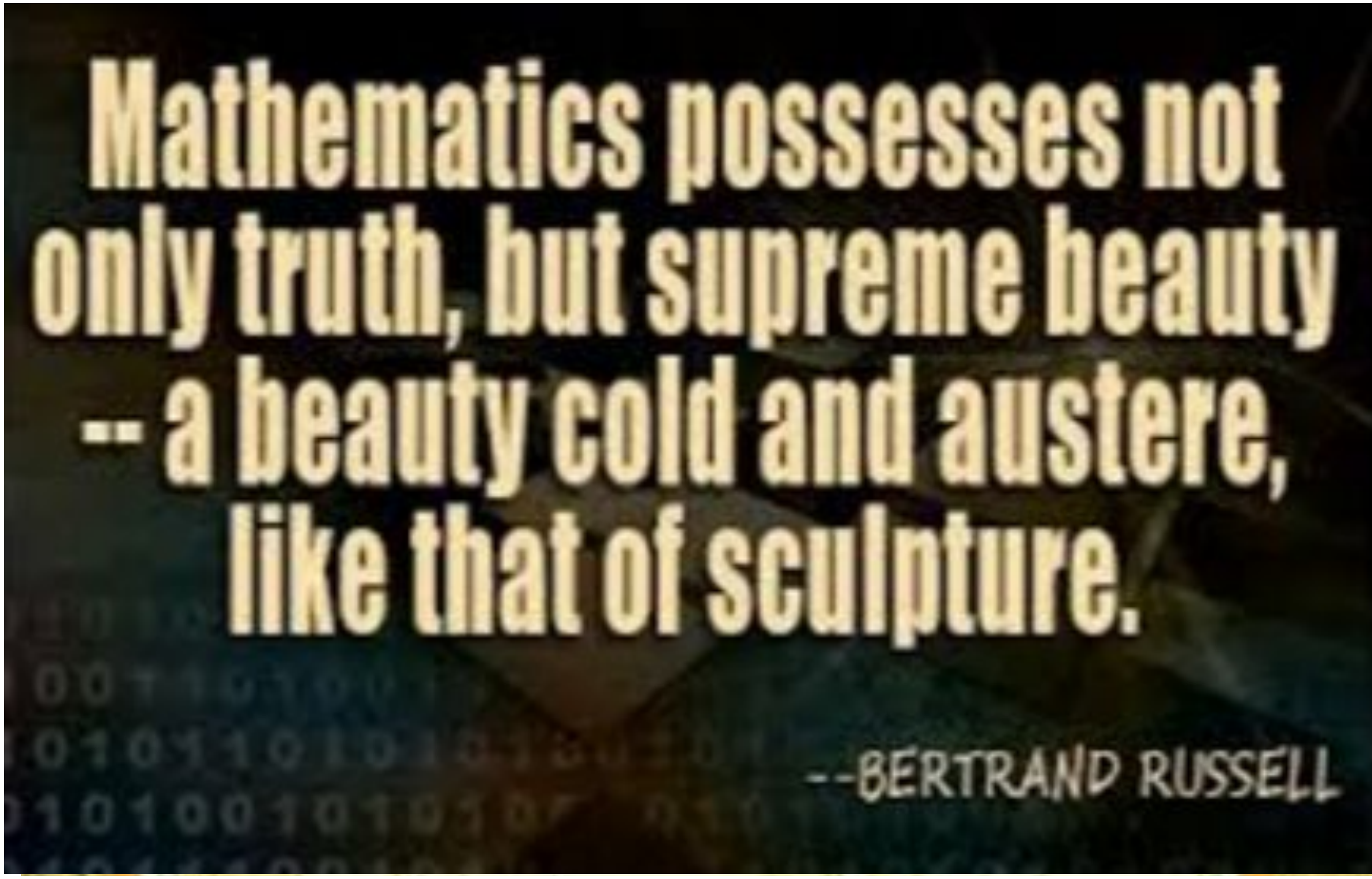
WEAPONS OF MATH DESTRUCTION

HOW BIG DATA INCREASES INEQUALITY AND THREATENS DEMOCRACY

CATHY O'NEIL

Mathematics / Computers

- Facts
- Objective
- Truth
- Logic

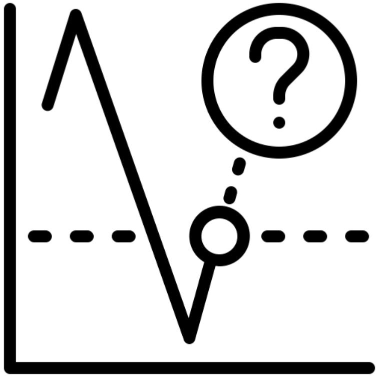


Mathematics possesses not only truth, but supreme beauty -- a beauty cold and austere, like that of sculpture.

--BERTRAND RUSSELL

Predicting Human Behavior

- Predict future events for individuals (e.g. future arrests, age of death, future loan default)
- Marketed as objective and fair
- Algorithms are nothing more than opinions embedded in code



What is a predictive algorithm?

- Data you train your algorithm on
  - Actual past meals, their attributes, and their success
- Data for current decision
  - Food I have
  - Time I have

What is a predictive algorithm?

- Criteria for success

How many **vegetables** did my daughter eat?
How much **whining** occurred?

How much **sugar** did she eat?

Why aren’t algorithms objective?

- Algorithms optimize for success.
  - Training data is **actual** past events
  - Current data may be **inaccurate**
  - The criteria for success is **not objective**.
- To understand how a specific AI system works, ask:
  - What are the limitations in data
    - previous past events
    - current data
  - What is the criteria for success?
  - Who benefits?
  - Who suffers?
- To understand how AI will change how we do things, ask:
  - What tasks **can be** assisted by generative AI?
  - What tasks **can not** be assisted by generative AI?
  - How does that change my field?

Weapon of Math Destruction

- Widespread
  - Affects a large *number* of people
  - The decisions *affect their lives substantially*
- Mysterious
  - The scoring algorithms’ formulas are not available
  - People may not even know they are being scored
- Destructive
  - Individual: They ruin people’s lives unfairly
  - Systemic: They make the original problem worse

Three Examples

- Teacher Assessment
- Hiring Practices
- Justice System

Teacher Assessment

- Two decade war on teachers
  - Must be the teachers' fault
  - Let's find the bad teachers, get rid of them

1st Gen: Student Scores

- Look for teachers teaching large number of students who did not pass standardized tests.
- What limitations do you see?

1st Gen: Student Scores

- Look for teachers teaching large number of students who did not pass standardized tests.
- Assumption: Score is entirely caused by teacher quality / performance.
- But poverty is correlated with performance.
- Therefore, you target teachers of students in poverty, not “bad” teachers

2nd Gen: “Value Added”

- Don’t blame teachers for students who start out behind
- Make a model for the expected score for each student in the class
- Teacher held accountable for ***difference*** between expected and actual score
- What limitations do you see?

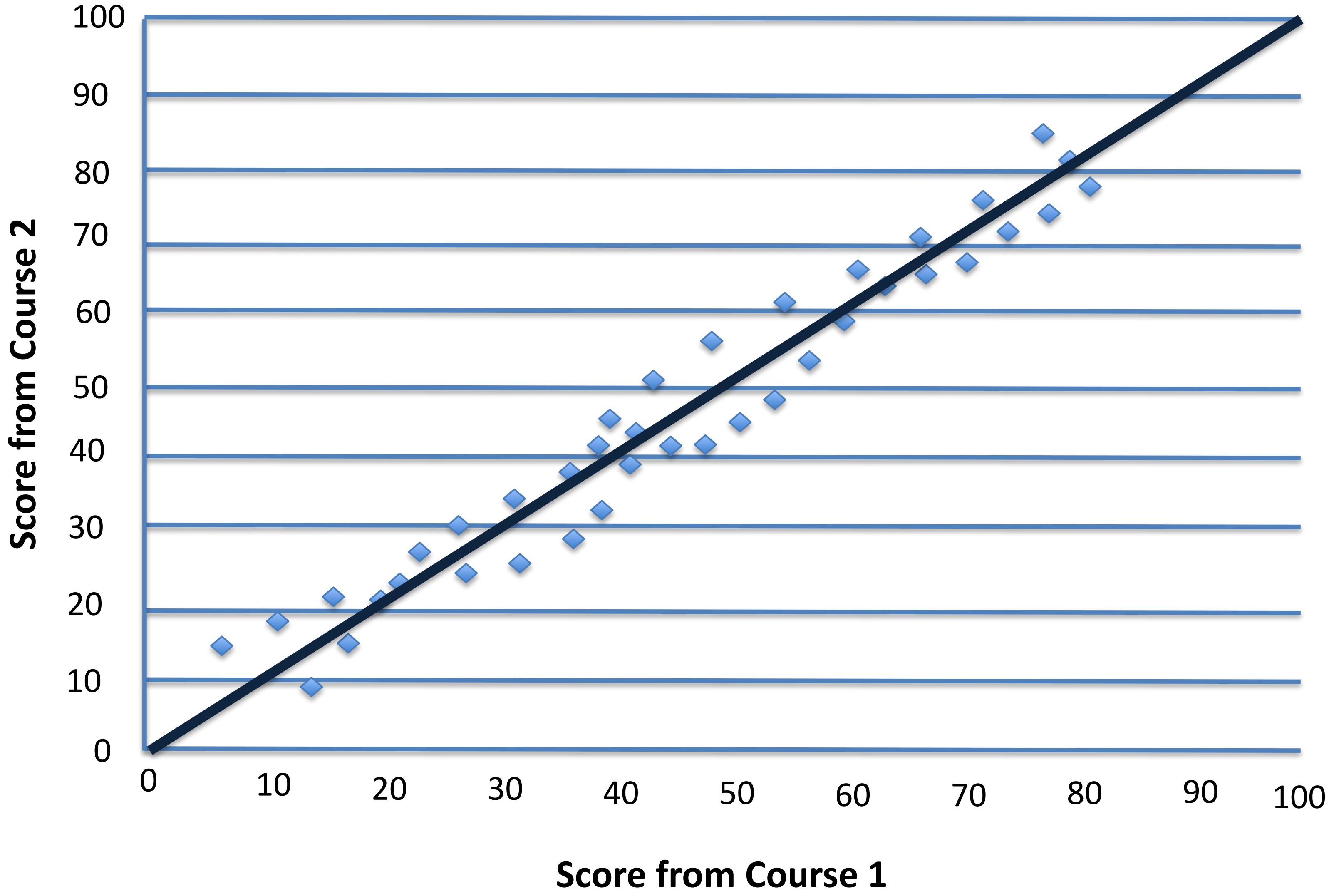
Limitations with “Value Added”

- If a teacher taught 20,000 students, this might work.
- Class of 35: Too small of a sample size
- It is a noisy model – test scores affected by temperature, time of day, medication, etc.
- Kids in poverty forget more in the summer.
- If a teacher cheats and makes student scores higher one year, the next teacher is punished.
- **Scores being used to deny teachers tenure**

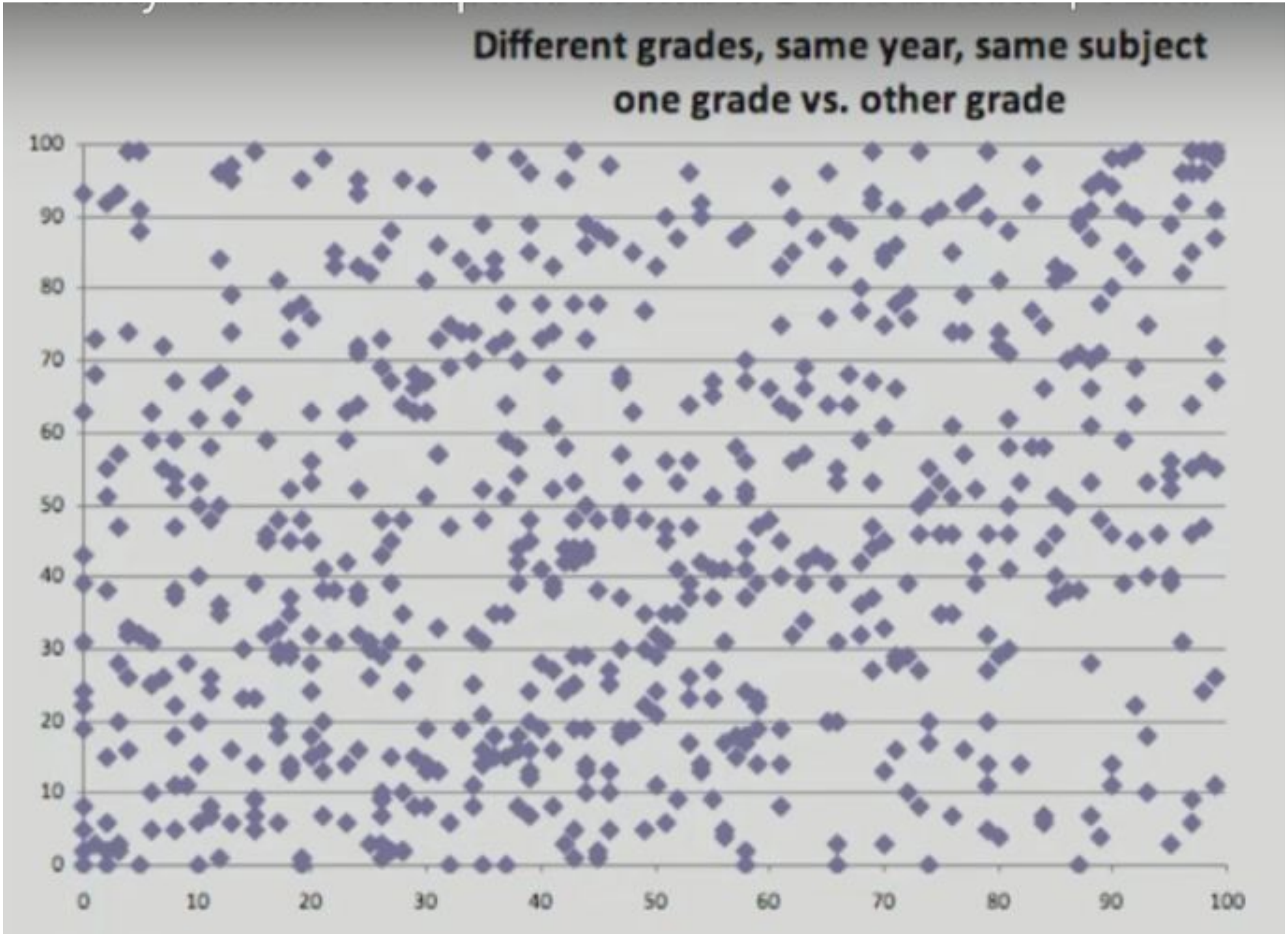
2nd Gen: Assessing Model

- How can we ***assess*** this model?
- Teachers who teach both 7th and 8th grade math had two scores.
- If this is a good model, then the same teacher should get a similar score in both classes.

Good Model Results



Actual Model Results



WMD

- Widespread
  - used in many urban districts nationwide
- Mysterious
  - algorithm is proprietary, FOI request was denied
- Destructive
  - Wrong individuals fired
  - Urban national teacher shortage: good teachers fired, quitting, retiring, not going into teaching because of perceived arbitrariness of evaluation

Using data / computer provides a façade of objectivity behind which bias hides

More Examples

- Parole system
- Life Insurance
- Auto Insurance
- Qualifying for a Mortgage
- Interest rates on Loans
- Voting Availability
- Targeted Marketing / Misinformation

Breakout #1

Consider one of the example systems

Identify

Biases in the data and their historical reasons

Purpose

Negative effect

- Parole system
- Life Insurance
- Auto Insurance
- Qualifying for a Mortgage
- Interest rates on Loans
- Voting Availability
- Targeted Marketing / Misinformation

Share out

Conclusions

- Data models are not objective – they reflect ***biases*** in ***historical data*** and ***definitions of success***
- Data scientists have an ethical obligation to consider **biases** in the models and how that affects ***individuals*** and the ***system***

Generative AI: Ethical Debate

Trains based on copyrighted material

Attempts to produce output that resembles training data

Plagiarism (artwork and text)

Breakout #2

Make ***genuine*** arguments for and against the following statement:

Generative AI should be allowed to use copyrighted work (for which it doesn’t have the copyright) for its training

What curbs / limits could be placed to be more ethical?

Arguments for

Humans also train on copyrighted data

Arguments against

AI has the potential to break copyright law by reciting an entire book - the user would be unaware of the level of plagiarism in a response

AI does not acknowledge the sources

Limits / Curbs

What are LLM’s?

Predictive models - nothing more!!

Given the question, they predict the first word of the response, then the second, then the third, etc.

They don’t “understand” the question

This is why code is so buggy!!!

Best for creative work, not for factual work!

Can GPT Help?
Supporting Teachers to Brainstorm Customized Instructional Scratch Projects

Minh Tran, David Gonzalez-Maldonado, Elaine Zhou, Diana Franklin
University of Chicago
ngminhtran@uchicago.edu

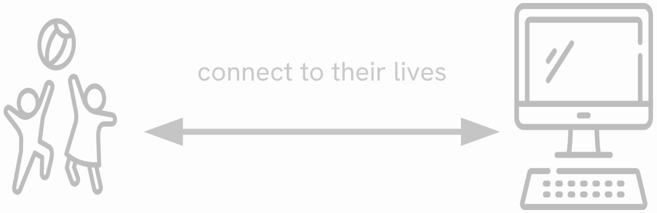
Motivation: LLMs for Class Preparation

- LLMs can support **college-level instructors** in class preparation in many ways
  - Creating high-quality materials at scale:
    - programming assignments (Jordan, 2024; Sarsa, 2022)
    - worked examples (Hassan et al., 2024; Jury et al., 2024)
    - code explanations (Leinonen, 2023; MacNeil, 2023)
  - Creating materials on demand by students:
    - personalized Parson puzzles (Hou et al., 2024)
    - contextualized programming assignments (Logacheva et al., 2024)

- Can LLMs support **K-8 instructors** in class preparation too?
  - Existing LLMs struggle to generate block-based code (Gonzalez-Maldonado et al., 2025)

Motivation: Customizing Teaching Materials

- Customizing available resources is a common practice for teachers in many subjects, including CS (Booth and Kellogg, 2015; Fincher et al., 2010)

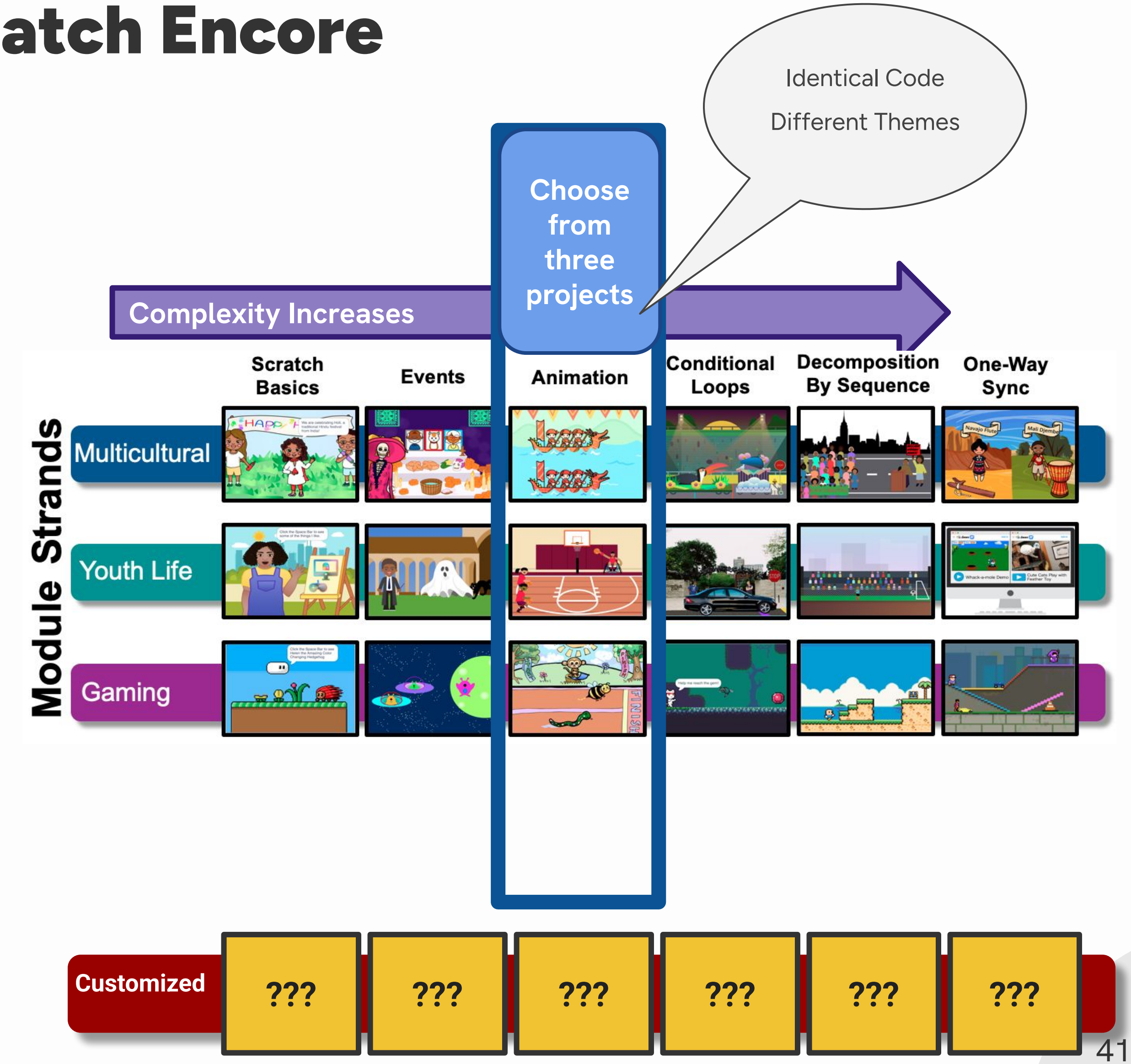


- But… customization can be **difficult** and **time-consuming**
  - Need a solid understanding of existing materials (Sosteric and Hesemeier, 2002)
  - Tension between a teacher’s time and professionalism (Tran et al., 2024)

Context: Scratch Encore

Strand/module structure

Within module structure (UMC)



Identical Code

Different Themes

Choose from three projects

Complexity Increases

Module Strands

Scratch Basics

Events

Animation

Conditional Loops

Decomposition By Sequence

One-Way Sync

Multicultural

Youth Life

Gaming

Customized

???

???

???

???

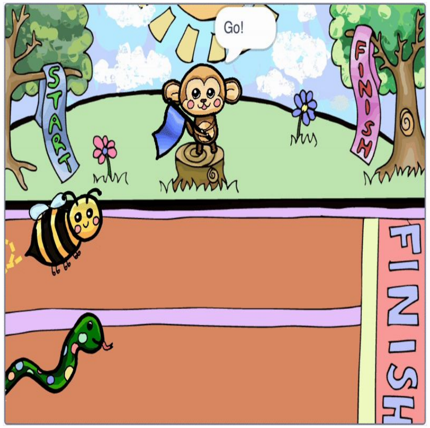
???

???

Our Approach

Using GPT to **mimic structured Scratch projects** from an existing CS curriculum and **situate the generated project ideas in different themes**

We need...



A base project to mimic

Tacos y Tamales Festival

Pokémon

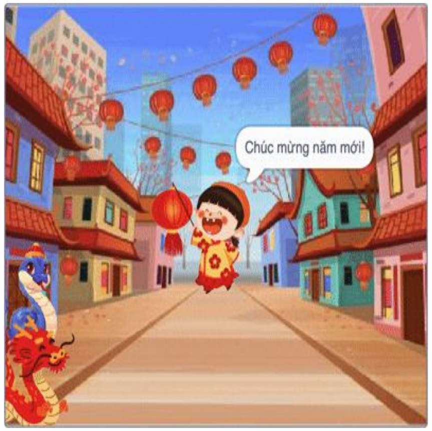
Color Run

Juneteenth

Lunar New Year

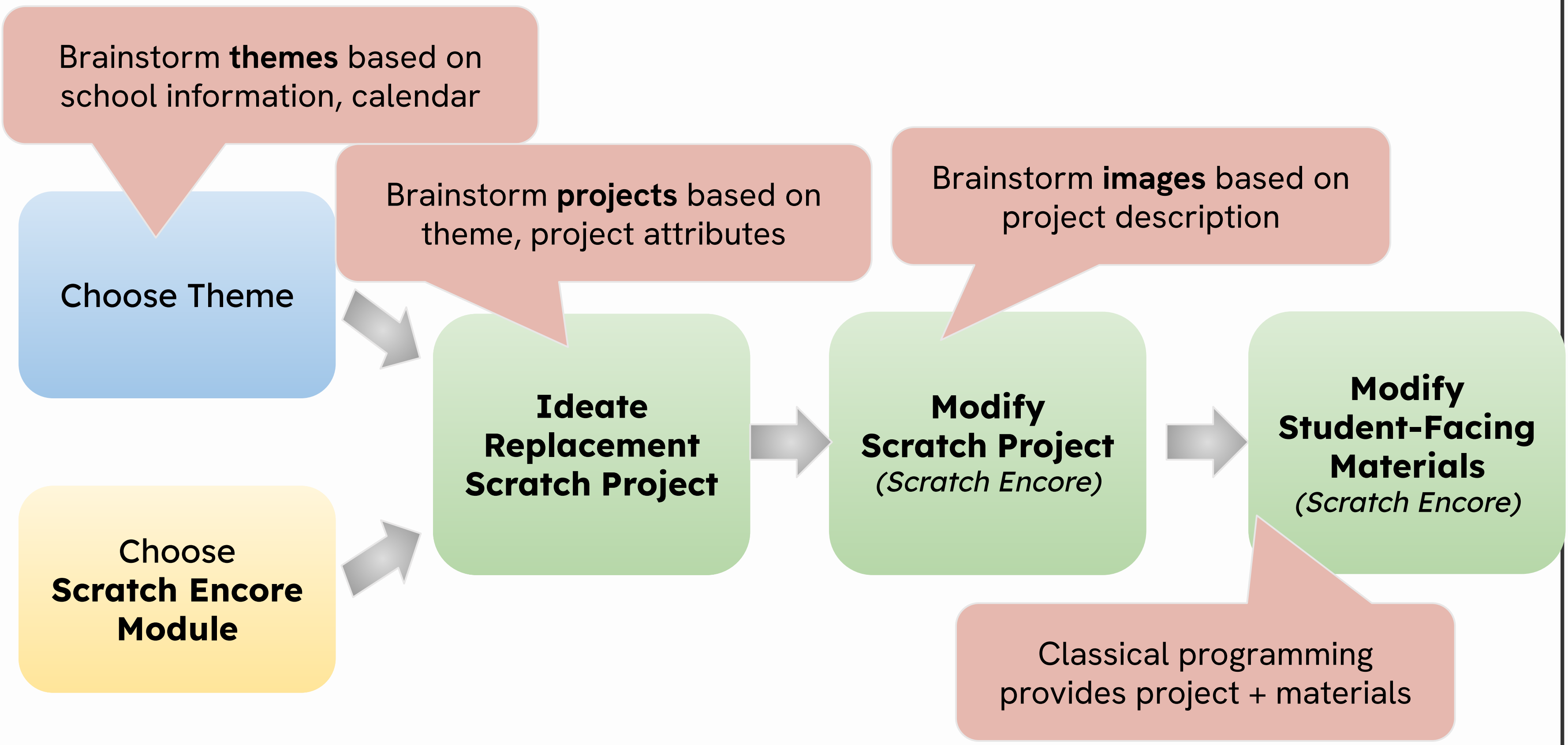
Cavaliers River Parade

A theme that students can connect to



Ideas for a custom project

Customization Process with GenAI



Brainstorm **themes** based on school information, calendar

Choose Theme

Choose **Scratch Encore Module**

Brainstorm **projects** based on theme, project attributes

Ideate Replacement Scratch Project

Brainstorm **images** based on project description

Modify Scratch Project *(Scratch Encore)*

Modify Student-Facing Materials *(Scratch Encore)*

Classical programming provides project + materials

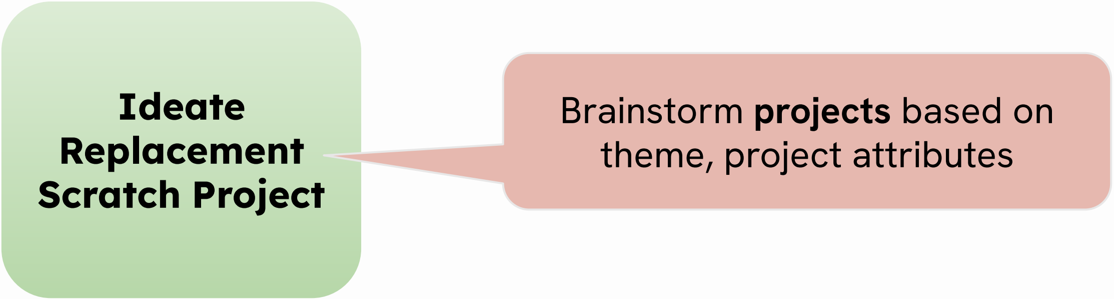
Research Questions related to Brainstorming a Project

RQ1

What are the strengths and weaknesses of GPT in generating customized, structured Scratch project ideas?

RQ2

To what extent can teachers use GPT to efficiently brainstorm ideas for customized instructional Scratch projects?



Ideate Replacement Scratch Project

Brainstorm **projects** based on theme, project attributes

Methods

Base Project Selection

Choose Modify projects from Scratch Encore (Franklin et al., 2020)

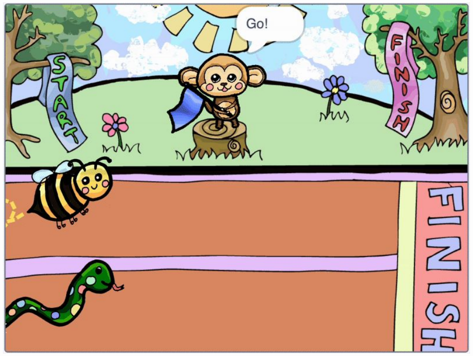
Project Idea Generation

Use GPT to generate 300 project ideas across 5 CS concepts and 20 themes

Project Idea Evaluation

Analyze GPT outputs for project theme quality, code alignment, multi-idea

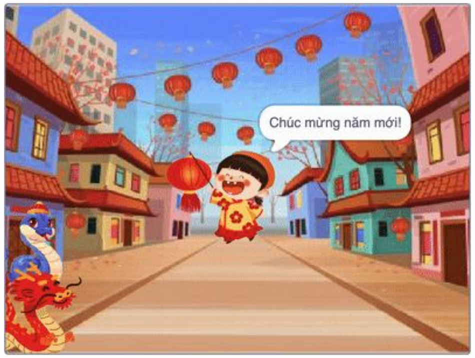
Scratch Encore Base Project: **Animal Race**



Backdrop: A track with a park in the background
Sprite: A monkey holding a racing flag, a bumblebee, a snake

Monkey waves a flag in place
Bumblebee and snake wiggle across the screen

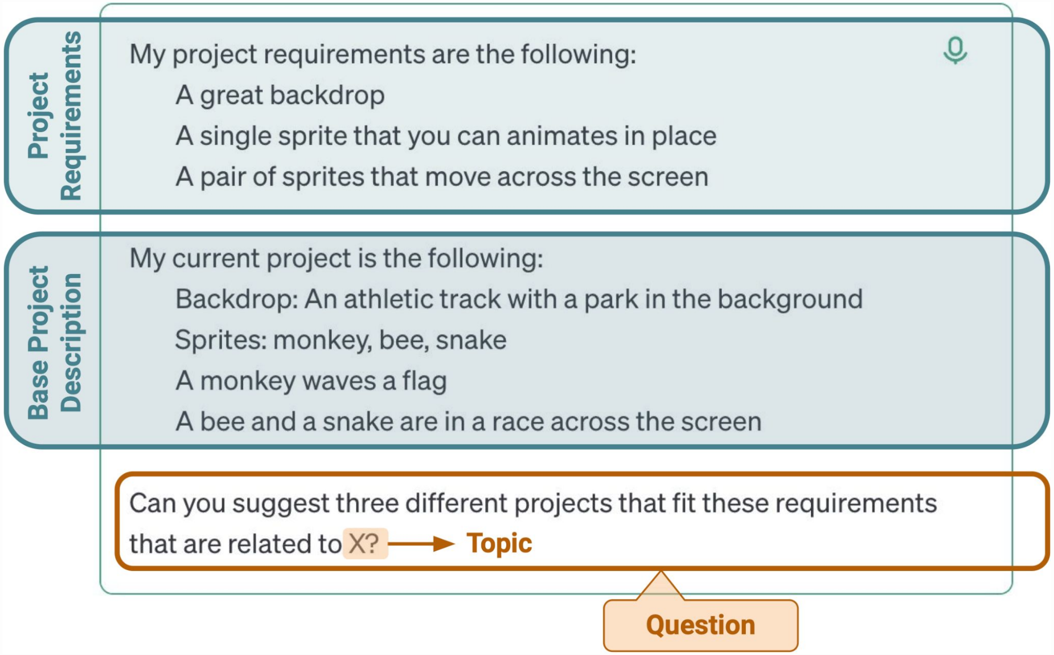
Custom Project: **Lunar New Year Animal Race**



Backdrop: A street with Tết decoration
Sprite: A girl holding a red lantern, a snake with a firecracker stick, a red dragon

The girl waving the lantern in place
Snake and dragon wiggles across the screen

Project Idea Generation



My project requirements are the following:

A great backdrop
A single sprite that you can animates in place
A pair of sprites that move across the screen

My current project is the following:

Backdrop: An athletic track with a park in the background
Sprites: monkey, bee, snake
A monkey waves a flag
A bee and a snake are in a race across the screen

Can you suggest three different projects that fit these requirements that are related to X?

- 5 SE modules
- 20 different themes
- 100 GPT queries

→ **300 GPT-generated project ideas in total**

Methods

Base Project Selection

Choose Modify projects from Scratch Encore (Franklin et al., 2020)

Project Idea Generation

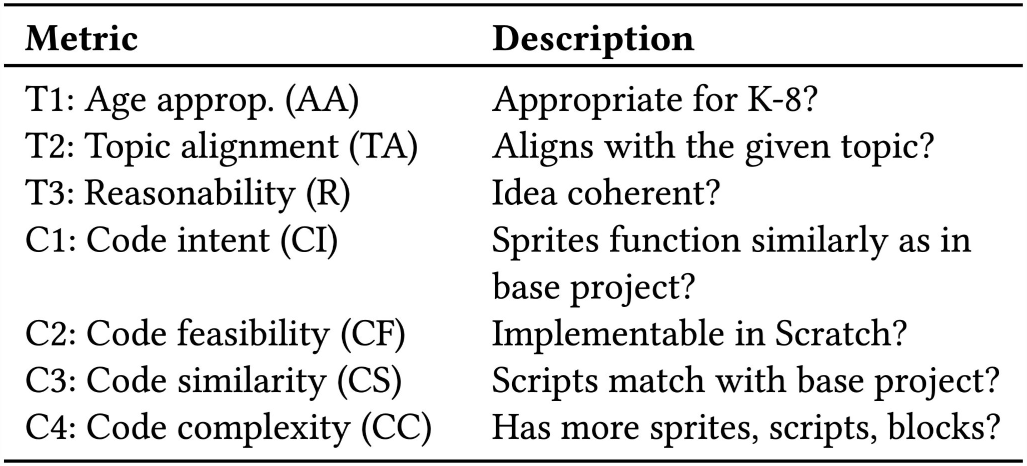
Use GPT to generate 300 project ideas across 5 CS concepts and 20 themes

Project Idea Evaluation

Analyze GPT outputs for project theme quality, code alignment, multi-idea

Project Idea Evaluation

- Evaluate each GPT-generated project ideas



Metric	Description
T1: Age approp. (AA)	Appropriate for K-8?
T2: Topic alignment (TA)	Aligns with the given topic?
T3: Reasonability (R)	Idea coherent?
C1: Code intent (CI)	Sprites function similarly as in base project?
C2: Code feasibility (CF)	Implementable in Scratch?
C3: Code similarity (CS)	Scripts match with base project?
C4: Code complexity (CC)	Has more sprites, scripts, blocks?

- Evaluate each GPT response
  - Variability (V): How different the three project ideas are from each other
  - Consistency (C): How many project ideas out of three satisfy all evaluation metrics

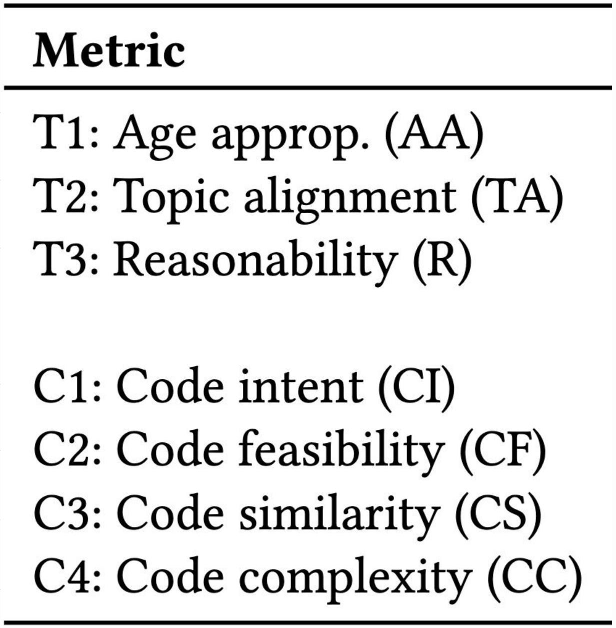
Example: No Revision

Theme: Food Festival

GPT-generated project idea

Backdrop: A food festival celebration with festival-goers in the background
Sprites: A chef, two waiters

- Chef slices vegetables with a single knife
- Two waiters carry trays of dishes across screen



	Metric
✓	T1: Age approp. (AA)
✓	T2: Topic alignment (TA)
✓	T3: Reasonability (R)
✓	C1: Code intent (CI)
✓	C2: Code feasibility (CF)
✓	C3: Code similarity (CS)
✓	C4: Code complexity (CC)

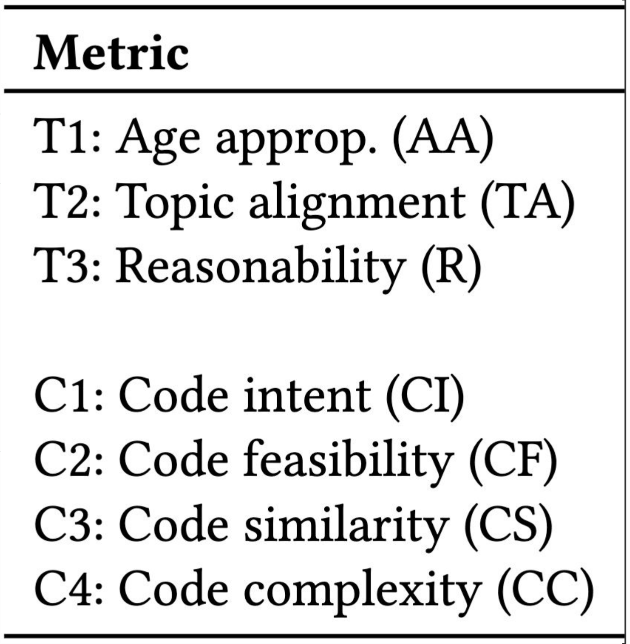
Example: Minor Revision

Theme: Aloha Festival

GPT-generated project idea

Backdrop: A tropical beach with coconut trees and a grass hut
Sprites: Hula hoop, tiki totem, flower headband

- Hula hoop moves around in a circle
- Tiki totem spins as it moves around
- Flower headband spins around a person's head as they move across screen



	Metric
✓	T1: Age approp. (AA)
✓	T2: Topic alignment (TA)
✗	T3: Reasonability (R)
✗	C1: Code intent (CI)
✓	C2: Code feasibility (CF)
✗	C3: Code similarity (CS)
✗	C4: Code complexity (CC)

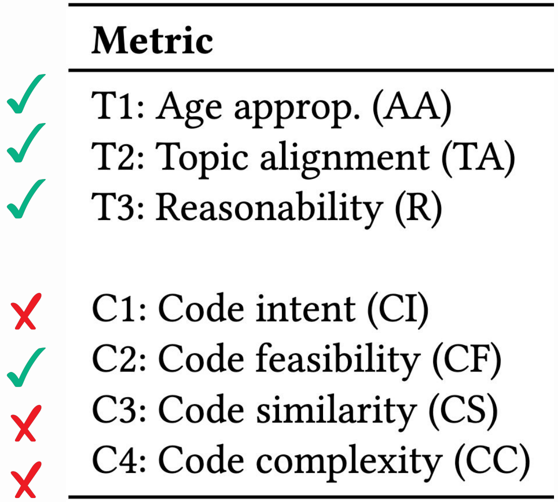
Example: Minor Revision (reasonability)

Theme: Aloha Festival

GPT-generated project idea (REVISED)

Backdrop: A tropical beach with coconut trees and a grass hut
Sprites: *A person hula hooping*, tiki totem, flower headband

- *A person hula hoops in place*
- Tiki totem spins as it moves around
- Flower headband spins around a person's head as they move across screen



	Metric
✓	T1: Age approp. (AA)
✓	T2: Topic alignment (TA)
✓	T3: Reasonability (R)
✗	C1: Code intent (CI)
✓	C2: Code feasibility (CF)
✗	C3: Code similarity (CS)
✗	C4: Code complexity (CC)

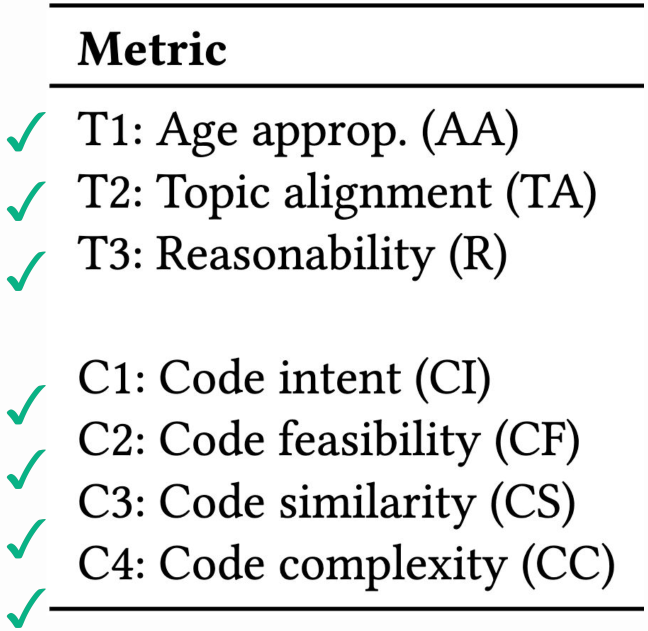
Example: Minor Revision (code)

Theme: Aloha Festival

GPT-generated project idea (REVISED)

Backdrop: A tropical beach with coconut trees and a grass hut
Sprites: *A person hula hooping*, tiki totem, *a person wearing a flower headband*

- *A person hula hoops in place*
- Tiki totem *spins across screen*
- *As the person moves across screen, the flower headband spins around their head*



	Metric
✓	T1: Age approp. (AA)
✓	T2: Topic alignment (TA)
✓	T3: Reasonability (R)
✓	C1: Code intent (CI)
✓	C2: Code feasibility (CF)
✓	C3: Code similarity (CS)
✓	C4: Code complexity (CC)

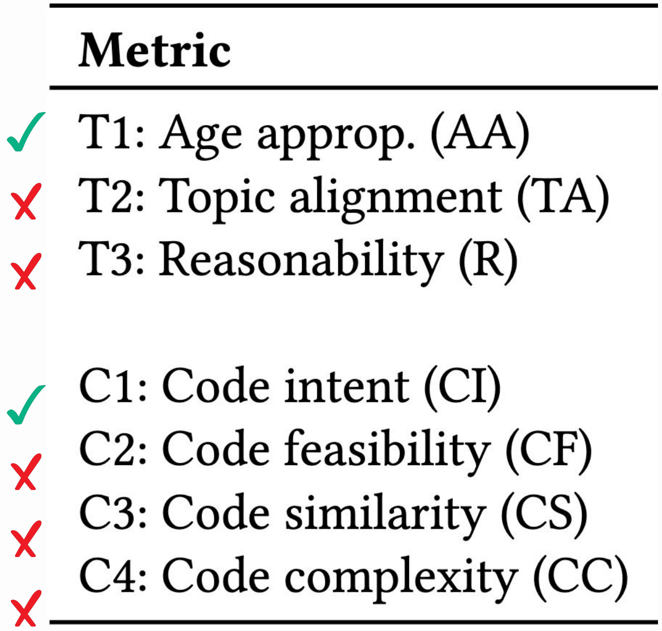
Example: Major Revision

Theme: Chinese Cuisine

GPT-generated project idea

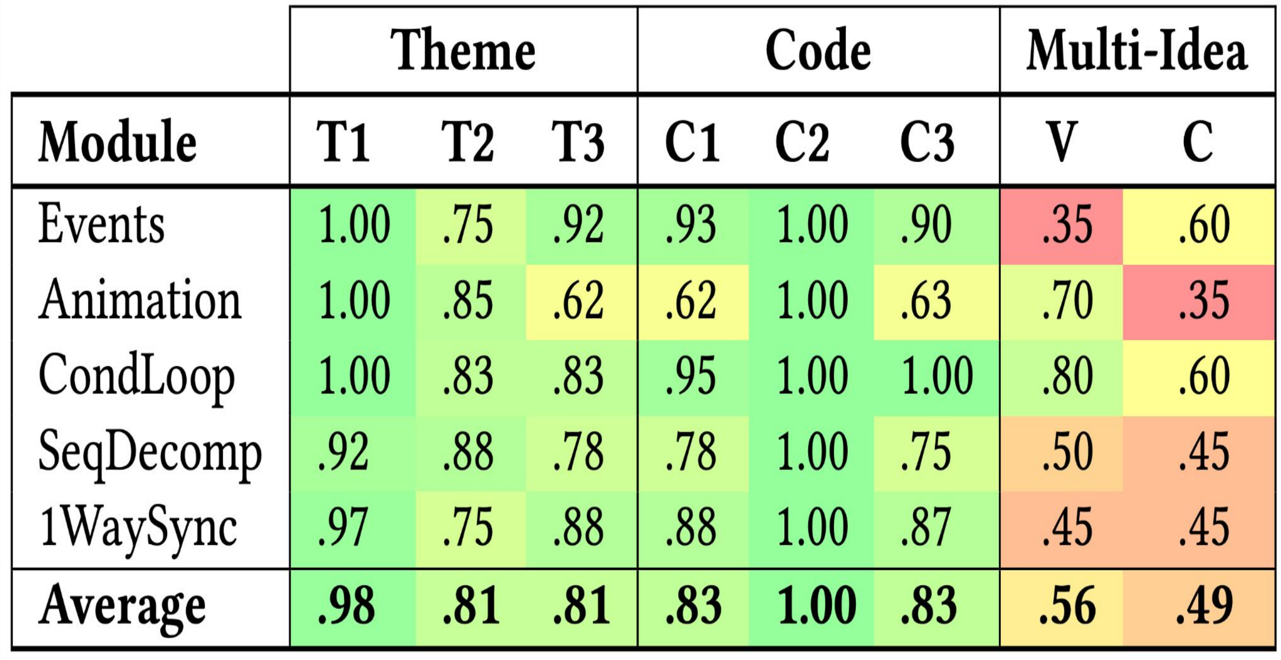
Backdrop: A Chinese city with busy streets and traditional architecture
Sprites: A cyclist, two cats

- Cyclist delivers dumplings across the screen
- Cats chasing each other around a delivery box



	Metric
✓	T1: Age approp. (AA)
✗	T2: Topic alignment (TA)
✗	T3: Reasonability (R)
✓	C1: Code intent (CI)
✗	C2: Code feasibility (CF)
✗	C3: Code similarity (CS)
✗	C4: Code complexity (CC)

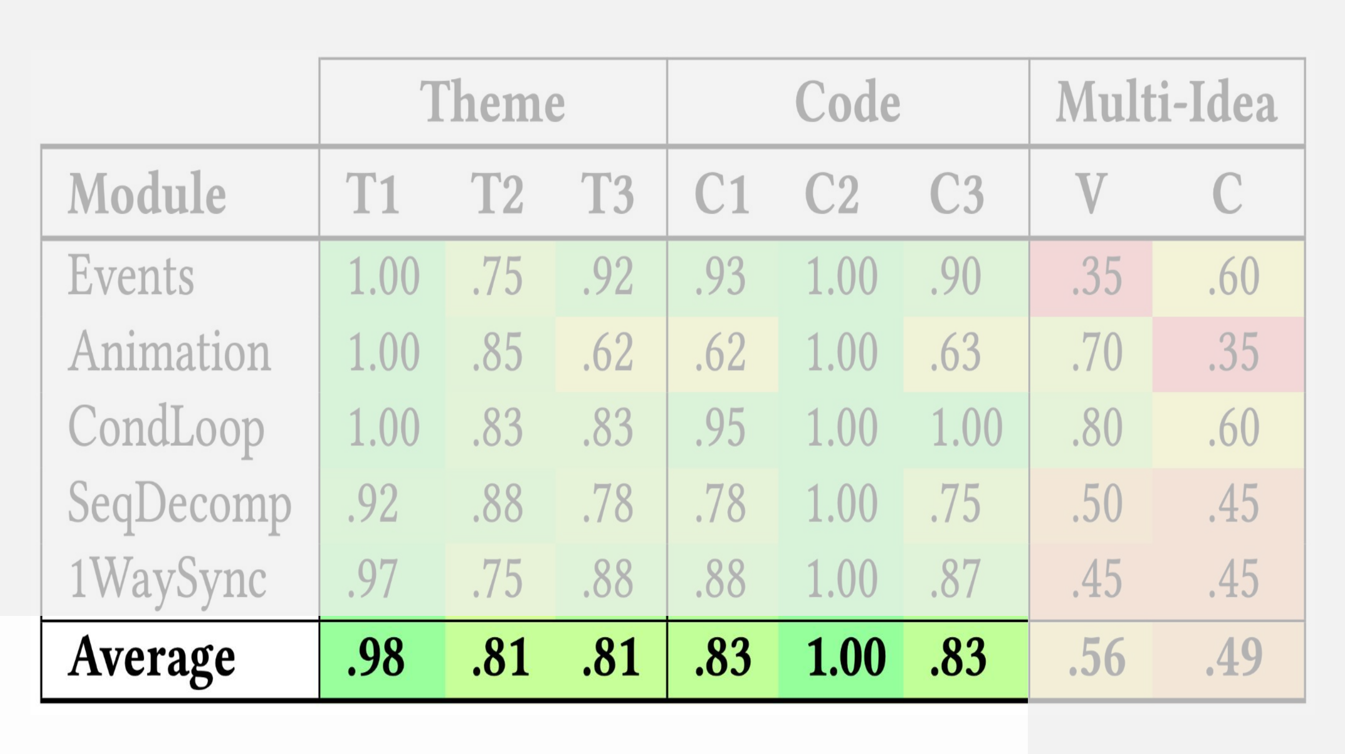
Results



	Theme			Code			Multi-Idea	
Module	**T1**	**T2**	**T3**	**C1**	**C2**	**C3**	**V**	**C**
Events	1.00	.75	.92	.93	1.00	.90	.35	.60
Animation	1.00	.85	.62	.62	1.00	.63	.70	.35
CondLoop	1.00	.83	.83	.95	1.00	1.00	.80	.60
SeqDecomp	.92	.88	.78	.78	1.00	.75	.50	.45
1WaySync	.97	.75	.88	.88	1.00	.87	.45	.45
Average	**.98**	**.81**	**.81**	**.83**	**1.00**	**.83**	**.56**	**.49**

Quality of GPT-generated project ideas across SE modules

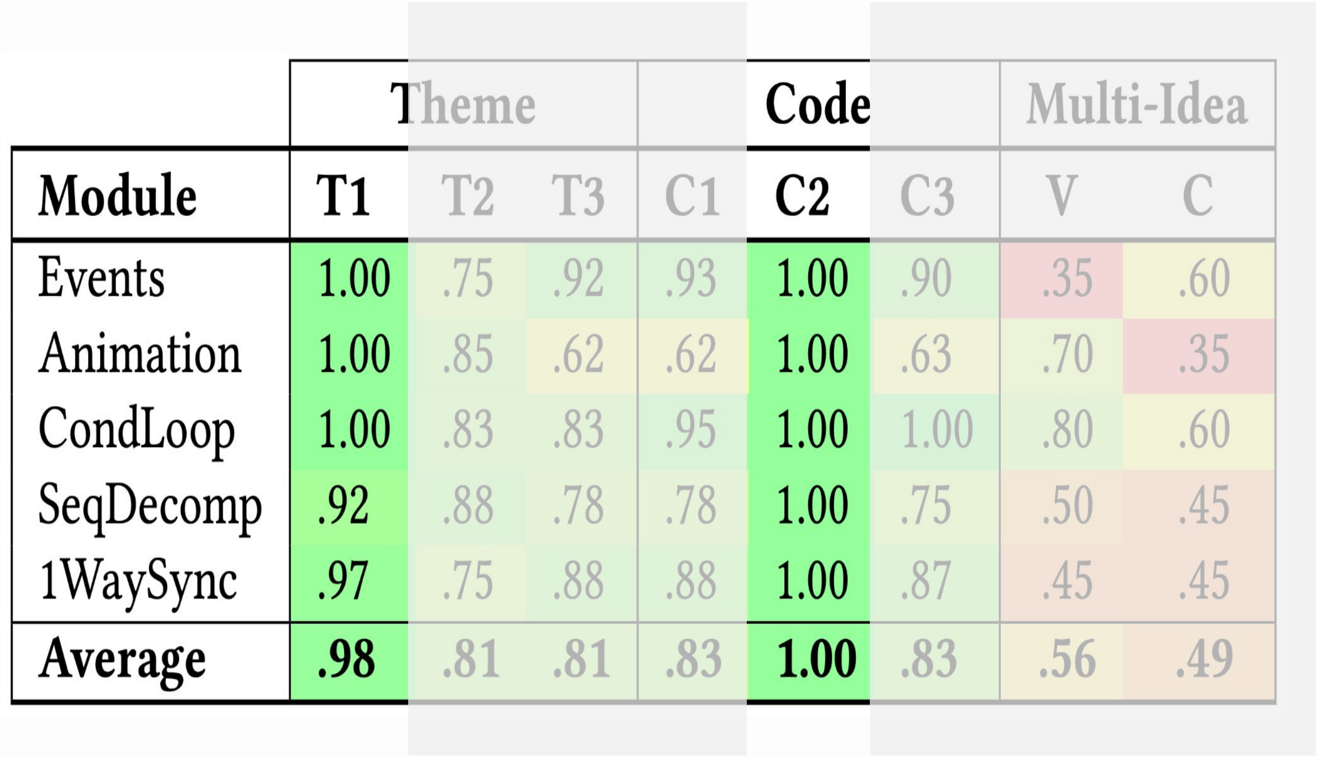
Results



	Theme			Code			Multi-Idea	
Module	T1	T2	T3	C1	C2	C3	V	C
Events	1.00	.75	.92	.93	1.00	.90	.35	.60
Animation	1.00	.85	.62	.62	1.00	.63	.70	.35
CondLoop	1.00	.83	.83	.95	1.00	1.00	.80	.60
SeqDecomp	.92	.88	.78	.78	1.00	.75	.50	.45
1WaySync	.97	.75	.88	.88	1.00	.87	.45	.45
Average	**.98**	**.81**	**.81**	**.83**	**1.00**	**.83**	.56	.49

On average, GPT-generated project ideas satisfy all individual criteria

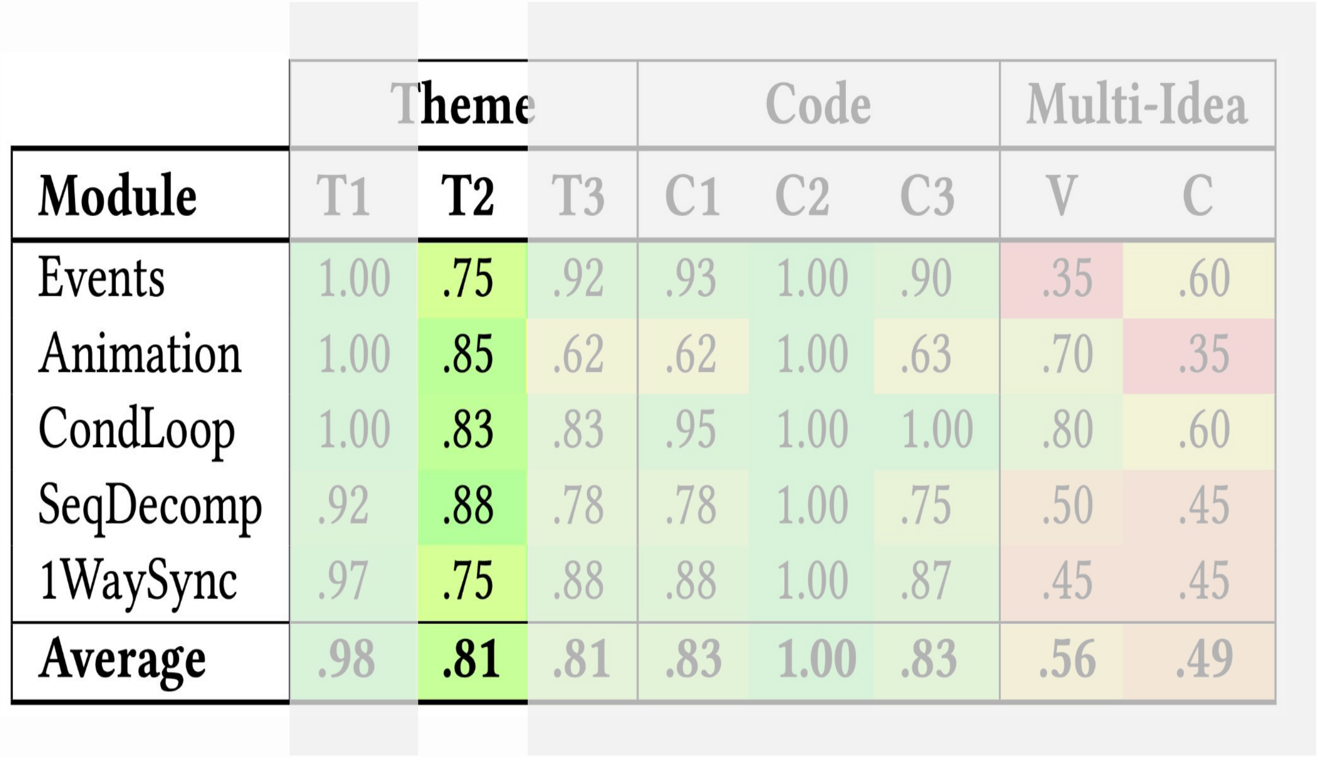
Results



	Theme			Code			Multi-Idea	
Module	**T1**	T2	T3	C1	**C2**	C3	V	C
Events	1.00	.75	.92	.93	1.00	.90	.35	.60
Animation	1.00	.85	.62	.62	1.00	.63	.70	.35
CondLoop	1.00	.83	.83	.95	1.00	1.00	.80	.60
SeqDecomp	.92	.88	.78	.78	1.00	.75	.50	.45
1WaySync	.97	.75	.88	.88	1.00	.87	.45	.45
Average	**.98**	.81	.81	.83	**1.00**	.83	.56	.49

Almost all GPT-generated project ideas are appropriate for K-8
and are implementable in Scratch

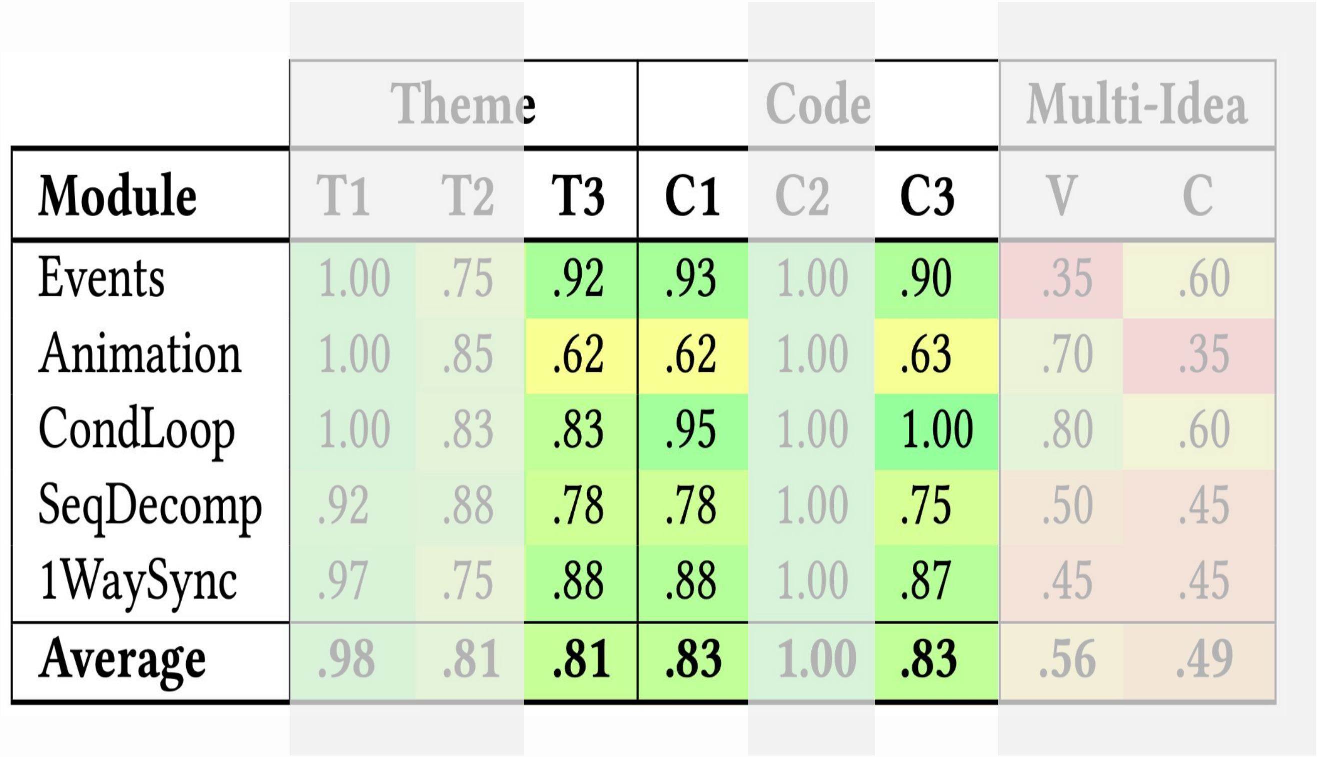
Results



	Theme			Code			Multi-Idea	
Module	T1	**T2**	T3	C1	C2	C3	V	C
Events	1.00	.75	.92	.93	1.00	.90	.35	.60
Animation	1.00	.85	.62	.62	1.00	.63	.70	.35
CondLoop	1.00	.83	.83	.95	1.00	1.00	.80	.60
SeqDecomp	.92	.88	.78	.78	1.00	.75	.50	.45
1WaySync	.97	.75	.88	.88	1.00	.87	.45	.45
Average	**.98**	**.81**	**.81**	**.83**	**1.00**	**.83**	**.56**	**.49**

GPT produces a majority of ideas that are aligned with the topic

Results

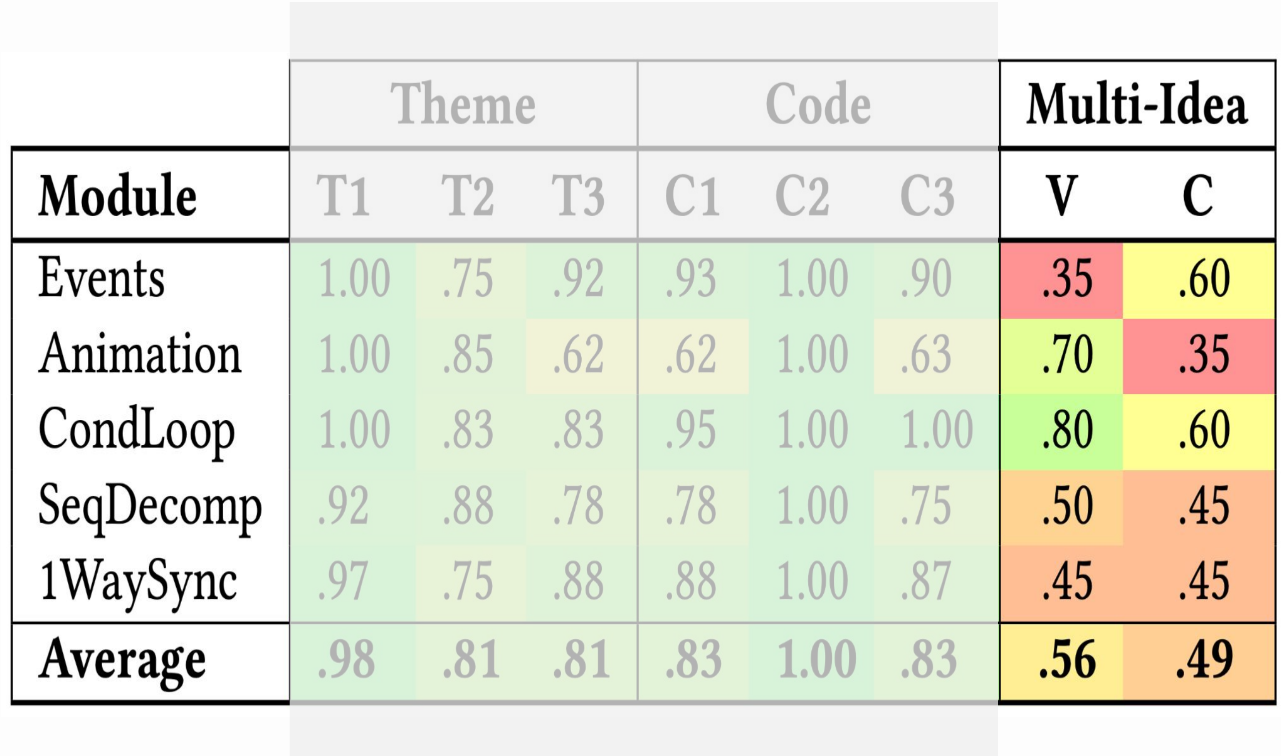


	Theme			Code			Multi-Idea	
Module	T1	T2	**T3**	**C1**	C2	**C3**	V	C
Events	1.00	.75	.92	.93	1.00	.90	.35	.60
Animation	1.00	.85	.62	.62	1.00	.63	.70	.35
CondLoop	1.00	.83	.83	.95	1.00	1.00	.80	.60
SeqDecomp	.92	.88	.78	.78	1.00	.75	.50	.45
1WaySync	.97	.75	.88	.88	1.00	.87	.45	.45
Average	.98	.81	**.81**	**.83**	1.00	**.83**	.56	.49

Reasonability, code intent, code similarity significantly varies by module

Results

Variable = Three generated ideas in an output tell clearly different stories
Consistent = At least two of the three generated ideas satisfy T1-T3 and C1-C3



	Theme			Code			Multi-Idea	
Module	T1	T2	T3	C1	C2	C3	**V**	**C**
Events	1.00	.75	.92	.93	1.00	.90	.35	.60
Animation	1.00	.85	.62	.62	1.00	.63	.70	.35
CondLoop	1.00	.83	.83	.95	1.00	1.00	.80	.60
SeqDecomp	.92	.88	.78	.78	1.00	.75	.50	.45
1WaySync	.97	.75	.88	.88	1.00	.87	.45	.45
Average	**.98**	**.81**	**.81**	**.83**	**1.00**	**.83**	**.56**	**.49**

Only 56 out of 100 GPT-generated project triplets are variable, and 49 of them are consistent in both theme quality and code alignment

Results

20% of the GPT-generated project ideas have complexity issues

- 63 potential issues identified:
    - Extra unique blocks (28)
    - Extra sprites (19)
    - Extra scripts (16)
- Common causes:
    - Complicated movements
    - Hidden sprites
    - Non-stationary sprites

RQ1: What are the strengths and weaknesses of GPT in generating customized, structured Scratch project ideas?

GPT can produce a large number of custom project ideas that are appropriate for students, situated in prompted themes, and well- or semi-aligned with technical requirements of prompted base projects

Complexity issues exist but (in many cases) can be solved by minor changes in project description

RQ2: To what extent can teachers use GPT to efficiently brainstorm ideas for customized instructional Scratch projects?

Teachers can use GPT with prompt only to brainstorm ideas for instructional Scratch projects, but may need additional scaffolds to learn how to spot and fix errors.

Breakout #3: Apply to Game Design

What do these strengths and weaknesses mean for using GenAI to help design a video game?

What would you want to brainstorm with it?

What “mistakes” do you need to watch out for?

Future Work

- **With additional scaffolds, can teachers filter out and adjust disqualified LLM-generated project ideas?**
    - Can teachers recognize project ideas that don't match?
    - How can we help them recognize, modify, and reject those project ideas?

Slides for Erica below here!!

LLMs and CS: How might coding change?

Before LLMs

After LLMs

LLMs and CS: How might coding change?

Before LLMs	**After LLMs**
Code from a blank screen	Coding from a buggy template
Code generators	Code Debuggers and Code Filler-inners

LLMs and CS: How might CS Ed change?

Before LLMs

After LLMs