Thinking With Agents

Webinar 2: Agentic AI for Teaching

Emily Beam & **Erkmen G. Aslim**

University of Vermont · Department of Economics

June 17, 2026 · Live Online

Welcome! How this works

Questions

Drop questions in the **Q & A** — we'll review as we go and raise them during Q & A time

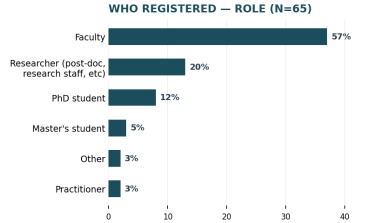
Recording

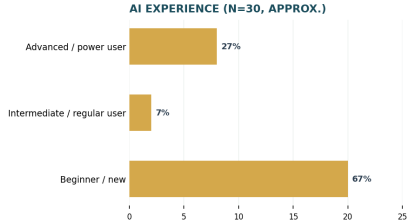
We're **not** posting the full recording publicly — but slides, skills, etc. will be posted on https://thinkingwithagents.github.io/

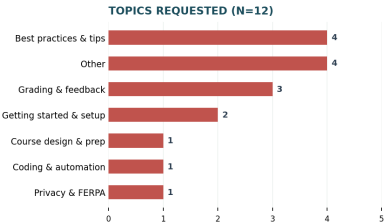
This is **mostly live**. Things may break — that's part of the honest picture.

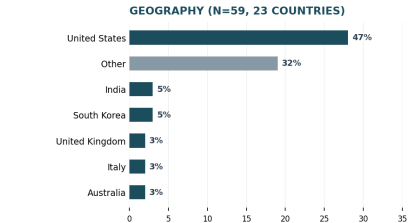
Who's in the room

Or at least, who signed up









Today's plan

Hour 1: Emily — grading

- **Agentic Overview**: what is even happening
- **Student data & AI:** what you can and can't feed a model
- **De-id in practice:** a live grading pipeline that never sees a student's name

Hour 2: Erkmen — teaching prep

- **Teaching prep is production** — agents make it cheap
- **Lecture Builder:** papers in $\rightarrow$ notes $+$ slides $+$ exams out
- **LMS dashboards:** one prompt $\rightarrow$ interactive tool your students use

Three ways to work with AI

1. Chat window

ChatGPT / Claude web

- ▶ Ask questions, paste text
- ▶ Can write code, make outputs
- ✗ Can't see your files

Like **texting a smart friend**.

2. In your editor

VS Code / Cursor

- ▶ AI inside your coding environment
- ▶ Sees the file you're in
- ▶ Autocomplete & inline edits

Like a **co-pilot** as you type.

3. Agent

Claude Code / Codex

- ✓ Reads & edits your files
- ✓ Runs commands
- ✓ Works across a project

Like an **RA sitting next to you** with unlimited caffeine.

The agent runs two ways: in your **terminal** (type `claude` / `codex`) or a **desktop app** — terminal gets new features first; the app is friendlier to start.

Today we're focusing on **agent-based**, because it's the biggest mindset/workflow shift.

A few terms we'll use

Term	What we mean
Model	The underlying AI system (e.g., Claude Sonnet, GPT-5)
Token	The unit AI models process — roughly $\frac{3}{4}$ of a word
Session	One conversation or working thread with the tool
Context window	The working memory of the current session
Agent	A tool carrying out a bounded task with some autonomy
Artifact	A saved output you can pull back up and reuse — a prompt, file, table, or memo

What do you need to get started?

- **A paid subscription**: $20/mo Claude Pro or ChatGPT Plus will be enough to get started!
  - In general, ChatGPT (currently) subscriptions yield more usage/$
  - Heavier users upgrade to $100/mo or $200/mo subscriptions
- **Terminal or AI desktop app (Claude or Codex)** — or both!

https://thinkingwithagents.github.io/install.html

Chatbot vs. agent

A chatbot

- You type a message, it replies
- You copy the output somewhere
- It has no idea what's on your computer
- Each turn starts fresh

An agent

- **Reads your actual files** — papers, code, data dictionaries
- **Runs commands** — Stata, Python, LaTeX, git
- **Remembers across turns** — standing instructions, project context
- Asks permission before doing anything risky

The agent's advantage isn't intelligence — it's **context**. It sees your files, your project structure, your conventions. That's what makes it useful for real teaching work, not just one-off questions.

What it can touch — and how you widen it

1. Default

Your working directory only.

Asks before each action. Where you start, and where most work happens.

2. Widen

Grant more as trust grows.

More folders, standing permissions for actions you've vetted — loosened deliberately.

3. Sandbox / VM

Isolated environment.

For big, autonomous jobs — it runs freely in a space that *can't* touch the rest of your machine.

Real isolation: run the agent in a sandbox using Docker, virtual environment, or other tools — free rein inside a container that can't touch your machine.

Part 1

Student Data and AI

What you can — and can't — feed a model

The 30-second version

Every country has rules about student data. The moment you put identifiable student work into an AI tool, those rules apply. The details vary — FERPA (US), GDPR (EU), POPIA (South Africa), PDPA (Asia-Pacific) — but the logic is the same everywhere.

✓ Path 1: Approved tool

Use an AI tool your **institution has vetted and contracted** — under institutional control, with a data-processing agreement.

✓ Path 2: De-identify

Properly de-identify the data first, so it's no longer protected information. Then use any tool.

⊘ The universal don't

Never paste identifiable student work or grades into a **personal/consumer** AI account. That's a disclosure your institution hasn't authorized.

US examples reference FERPA (34 CFR Part 99); principles generalize across frameworks. Not legal advice — defer to your institution's policy.

What counts as identifiable student data

Student data protection laws cover...

- Grades, transcripts, class lists, student work
- Records **held by a vendor on your behalf** (they stay protected after handoff)
- **Direct** identifiers (name, ID, email)
- **Indirect** identifiers (DOB, writing style, topic)

The test: could a reasonable person in your community identify the student? If yes, it's identifiable.

The mosaic effect

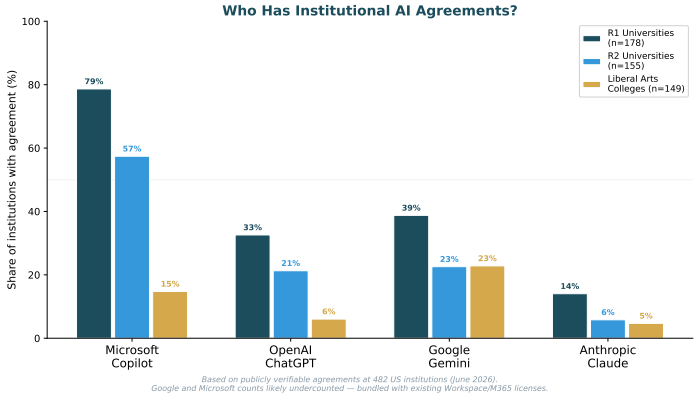
In a **small seminar**, an "anonymized" essay can still be re-identifiable from:

- topic choice
- writing style
- facts you already know about the student

De-identification means addressing **combinations** — not just deleting the name.

A *student* pasting their *own* work into a consumer tool is their choice. The obligation runs to **you and the institution**.

What Path 1 actually looks like across US institutions



If your institution hasn't contracted an AI tool, **Path 2 (de-identify) is your only clean option.**

Concrete classroom scenarios

Scenario	Verdict	Why
Named essay into consumer ChatGPT for feedback	✗	Unconsented disclosure
Same essay, **fully de-identified**, low re-ID risk, general tool	⚠	Defensible — your judgment
Same essay, in the **university's contracted** tool	✓	Vendor is a school official
Upload the class gradebook to summarize performance	✗	Grades are PII
Course chatbot **trained on** student submissions	✗	Needs approval + DPA
AI to draft a **generic** rubric or lecture (no student data)	✓	No education records involved

The dividing line is almost always: **identifiable student data + a tool that isn't institutionally controlled.**

Part 2

De-Identification in Practice

A live grading pipeline that never sees a student's name

The assignment I want to grade

Economic Delights 4: Traffic in India

Assignment

Due June 9 at 11:59 PM

Before you start this assignment, **make sure you've read Chapter 10!**

Then, answer the following questions. Your answer should total at least **350 words**

1. What market failure is caused by excessive horn honking? Can it be solved through Coase's theorem? Why or why not?

2. Then, watch this video (also below) about solving the problem of excessive horn honking in India. How do the Mumbai Police internalize the externality?

3. Would the Mumbai Police's approach work in a city like NYC? Why or why not?

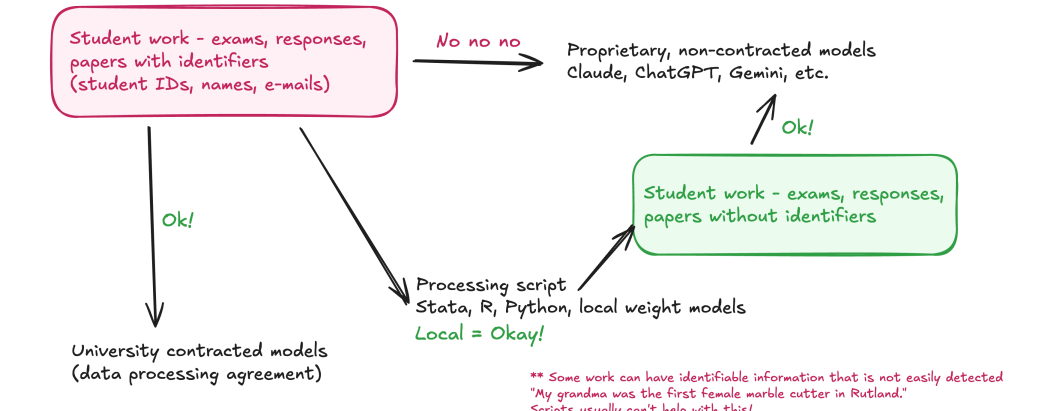
Grading:

2 points: Thoughtful and complete answers, meeting full word count

1 point: Partial answers, at least 150 words provided

0 point: Woefully incomplete answers, less than 150 words

The decision flow: student work → AI



Some student work has identifiable *content* that isn't a named entity — e.g., "My grandma was the first female marble cutter in Rutland." Scripts strip names and IDs; they can't catch this. For these cases, route to a contracted tool instead.

A grading pipeline

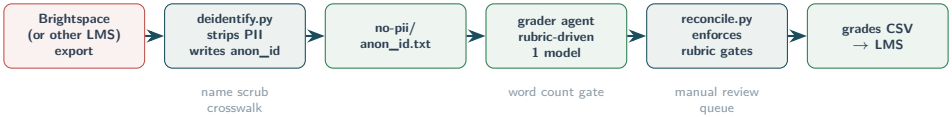
Path 2 says: **de-identify first, then use any tool you want.**

What does that actually look like for the most common task — **grading essays at scale**?

What a pipeline enables

- **Strip PII automatically** before any AI sees the work
- Run a **rubric-driven agent** on anonymous text
- **Your review** catches edge cases and flags
- Export a **grade CSV** straight to your LMS

The pipeline



At no point does the AI see a name, student ID, or anything that re-identifies the student.

PII stays locked

Filenames → crosswalk → `anon_id`.
Student names scrubbed from text.
`pii/` folder never sent to agent.

Any submission format

Handles HTML text-box, DOCX, and PDF — all three formats Brightspace produces.

My PII guardrail: what I actually use

I asked the agent to write a **pre-tool hook** — a shell script that fires automatically before every file read or terminal command. It took about 30 seconds. I've been refining it since.

What it blocks

- **Read tool**: data/geo files (`.csv`, `.dta`, `.sav`, `.shp`, `.parquet`, etc.)
- **Bash dumps**: `cat`, `head`, `grep`, `awk` on data files
- **Protected folders**: paths where even non-data files contain PII
- **LMS exports**: detects Brightspace/Canvas folder-name patterns where student names are baked into subfolder titles

What it allows

- `python3 script.py` — scripts read PII, write outputs to disk, print only **aggregates**
- **Schema inspection**: column names and types, no values
- **Per-project unlock**: add a folder to `SAFE_PATHS` to mark it PII-free

The principle: a script may read PII; the agent context must never see raw PII rows.

Live Demo

3 anonymized submissions · economic-delights rubric

deidentify → count words → grade → reconcile → CSV

What the agent sees — and what stays yours

The agent receives

```
anon_id:  STU04
Question 1:  Describe the economic tradeoff...
Student response:  The opportunity cost...[391
words]
```

No name. No student ID. No course section.

The agent returns

- **Score** (0 / 1 / 2 per rubric)
- **Justification** (1–2 sentences)
- **Review queue**: flag if borderline

Your judgment

- **Edge cases**: embedded links, borderline word counts
- **Rubric calls**: you set the rubric, review anything the explanation doesn't support
- **Final sign-off**: grade CSV is yours before upload

The pipeline handles **mechanical work at scale**. **Pedagogical judgment** — what counts as thoughtful, what warrants a flag — is **still yours**.

Built once, used every semester

To reuse for the next assignment

1. Copy the scripts to the new assignment's grading folder
2. Update the **rubric YAML** (questions, thresholds, score levels)
3. Point `DOWNLOAD_DIR` at the new Brightspace export
4. Run: `deid` → `count_words` → `grade` → `reconcile`

This session's run: ED4 (Coase theorem), 18 submissions. 16 auto-graded in ∼4 min. 1 manual flag, 1 embedded-doc gotcha.

Don't want to build it? Describe your rubric and export format to the agent — it writes the scripts for you.

Extensions: from grading to better feedback

The same pipeline that grades can also **improve what students get back**.

- **Literature connections**: run a search for sources related to each student's topic — suggest 2–3 papers they should read next
- **Constructive feedback**: generate a paragraph of specific, rubric-anchored feedback for each submission (not just a score)
- **Harmonized passes**: run a second agent pass across all graded submissions to flag inconsistencies in scoring
- **Smarter de-identification**: open-weight PII models (e.g., OpenAI Privacy Filter, ∼1.5B params, Apache 2.0) run **entirely on your laptop** — learned entity detection instead of regex, ∼96% F1

Each of these is a separate agent step you can add to the pipeline — or skip. The base pipeline works without them.

Over to Erkmen

This hour in one line: know the rule (FERPA) · use the path (de-identify) · let the agent do the mechanical work · you own the judgment.

Erkmen picks up here: teaching prep, the lecture-builder, and an interactive tool your students use inside your LMS.

Reference: the three enterprise tools, compared

	ChatGPT Edu / Enterprise	MS 365 Copilot	Claude Edu / Enterprise
Trains on your data by default?	**No**	**No**	**No**
Consumer tier trains by default?	Yes (opt-out)	N/A	**Yes** since Aug 2025
Explicit FERPA "school official" language?	No	**Yes** (99.33(a))	No (processor + DPA)
Data Processing Addendum?	Yes	Yes (OST DPA)	Yes (auto-incl.)
Retention	Configurable	Configurable	Until deleted (API: 30d)
Zero-data-retention option?	Via API	—	**API only, not chat**

All entries are **vendor representations**, not adjudicated FERPA compliance — terms change fast. **Microsoft** alone publishes an explicit FERPA school-official commitment; the others rely on the contract/DPA + institutional vetting. *Sources: openai.com, learn.microsoft.com, privacy.claude.com — as of June 2026.*

What the Dept of Education has — and hasn't — said

No AI-specific rules

No AI-specific FERPA regulations exist (as of June 2026). The existing **technology-neutral** guidance governs AI vendors directly.

The framework (school-official exception, direct control, data-minimization) is tech-neutral — AI tools are just a subset of "online educational services."

The operative ED documents:

- Third-Party Service Providers under FERPA (2015)
- Online Educational Services: Requirements & Best Practices (2014)
- **Model Terms of Service checklist (2016)** — ready-made for vetting a tool's ToS

studentprivacy.ed.gov

So how worried should you be? The enforcement reality

How FERPA is enforced

- **No private right of action** — a student cannot sue you under FERPA *(Gonzaga v. Doe, 2002)*.
- ED's ultimate stick is **withdrawing federal funds** — never actually used.
- Real ed-tech enforcement has run through **COPPA/FTC**, not FERPA (Edmodo, $6M, 2023).

The real exposure is institutional and reputational, not a lawsuit.

That's exactly why your university **vets and contracts** tools centrally — and why the safe move is to **use the approved one**.

ED's only recent vendor action (the May 2026 Canvas/Instructure letter) was about a **data breach** — not AI. There is still **no AI-specific FERPA guidance**.

The question everyone asks: “does it train on my data?”

The FERPA answer

Under the school-official exception, a vendor may use PII **only for the contracted purpose** and may not reuse it for its own ends.

So training a vendor's models on **identifiable** student records is an **unauthorized reuse** — unless you authorize it.

De-identified data is outside FERPA — training on it is fine.

ED 2014/2016 guidance; CRS R46799. A sound inference from the use-limitation rule — ED has no AI-specific ruling.

The vendor reality

- OpenAI, MS 365 Copilot, and Claude (**commercial/edu** tiers) all state they don't train on your prompts *by default*.
- These are **vendor statements**, not FERPA rulings — and consumer tiers are excluded.
- Only **Microsoft** publishes an explicit FERPA “school-official” commitment.

Full three-way comparison on the reference slide.