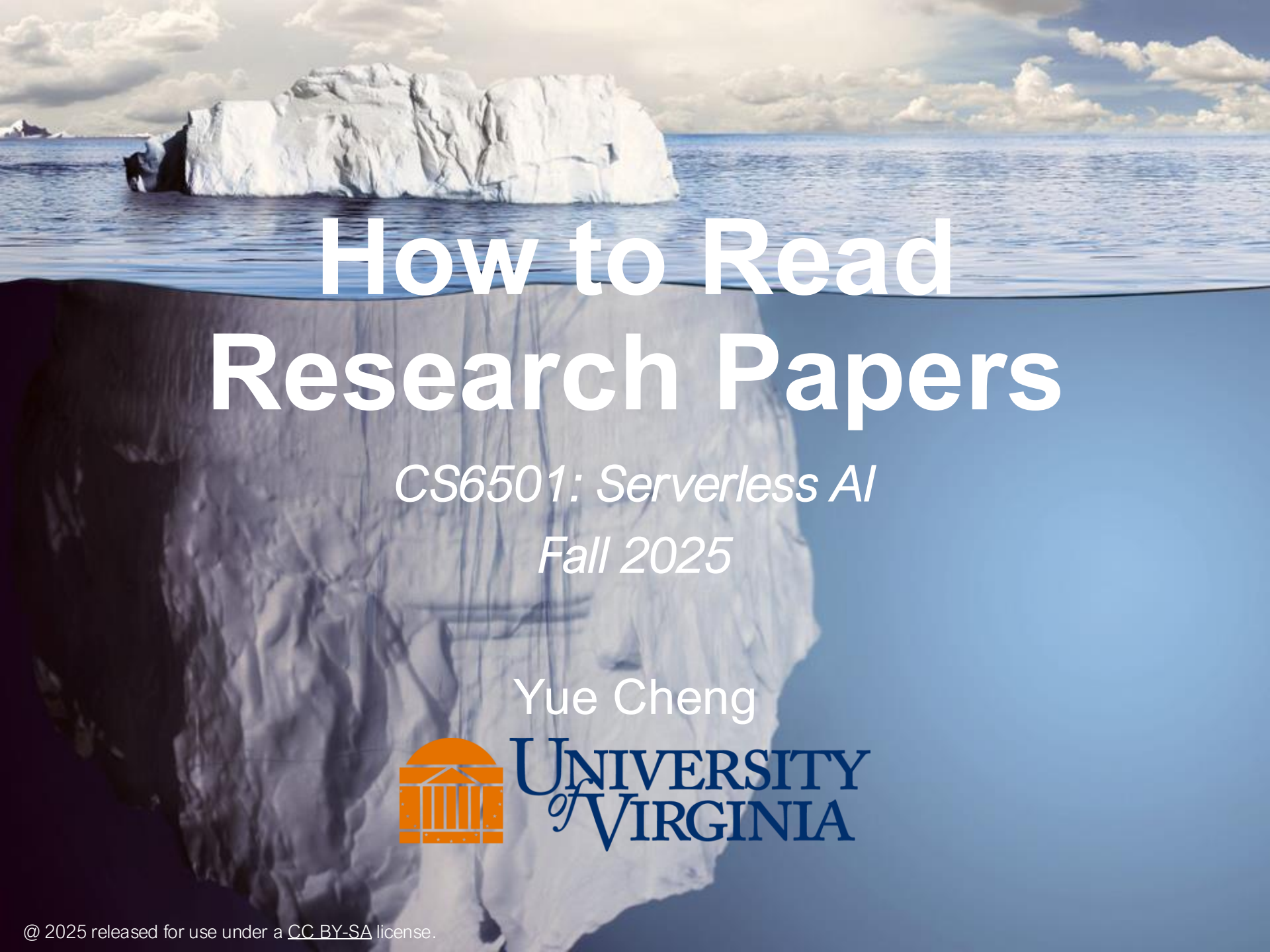

How to Read Research Papers

CS6501: Serverless AI

Fall 2025

Yue Cheng

University of Virginia

How to read a (systems) research paper?

- Paper reading is an important skill in grad school
  - To learn how to efficiently and effectively read a **systems** research paper
  - Might be slow at the very beginning
  - But you will get better the more you practice reading

- We will read dozens of papers in thorough detail
  - Impossible to innovate if you don't know the literature well

#1 rule:

Do NOT worship any paper or author

- A paper is not a "truth" but an "opinion" (or claim)
  - You should have your own judgement

- Critical thinking is a must in grad school
  - Papers are arguments based on authors' work / perspective
  - You are welcome to reject the arguments, criticize the approaches, and question the results
  - However, you will need to backup your criticisms and rejections

How to read a (systems) research paper? (Griswold's version)

1. What are the motivations?
2. What is the proposed solution?
3. What is the work's evaluation of the proposed solution?
4. What is your analysis of the identified problem, idea, and evaluation?
5. What are the contributions?
6. What are future directions for this research?
7. What questions are you left with?
8. What is your takeaway message from this paper?

https://cseweb.ucsd.edu/~wgg/CSE210/howtoread.html

How to read a (systems) research paper? (Cheng's version)

1. What problem does this paper tackle?
   - Why is it important or why do you think it is not as important (do you disagree)?
2. What is the solution design & implementation?
   - What is the limitation of the solution?
3. What do you learn from this paper?
4. Does it resonate your experience or benefit your own research or work?
   - If so, how do you want to use their solution?
   - If not, why doesn't the paper interest you? (This might help identify/clarify your interest.)

How to read a (systems) research paper? (Cheng's version)

1. What problem does this paper tackle?
2. What is the solution design & implementation?
3. What do you learn from this paper?
4. Does it resonate your experience or benefit your own research or work?

Use AI as your best tutor, w/ cautions

- AI cannot replace you → Don't simply throw papers at AI, there's **NO** substitute for first-hand reading ← **W/o reading yourself, you won't even know what to ask!**
- Use AI to clarify & reinforce → Ask it to better explain, rephrase, or summarize → Keep asking until you understand it
- **For advanced learners:** Leverage AI to reproduce the problem or build small demos ← Real understanding comes from **DIY**

Example prompt to get started

You are an experienced systems researcher. You are critical, clear, and pretty good at teaching & communicating ideas in an intuitive way.

I am a senior undergrad [or first-year PhD] student in CS. I don't have first-hand experience with KVM, Lambda, serverless computing, or hardware virtualization, but I am genuinely interested in learning more about state-of-the-art research in cloud and serverless computing systems. Do not assume I have prior background in any of these techniques.

Now, read the paper from the attached URL [or PDF]. Answer my questions. [followed by your questions…]

[Why Firecracker is built atop KVM, what is exactly QEMU doing, and why it is not built directly atop QEMU.]

Example prompt to get started (cont.)

Just as many MapReduce learning materials distill the original paper, summarize the Firecracker paper into a concise, intuitive format. Think of it as a “Firecracker Essentials Guide” so I can quickly grasp its essence without getting bogged down in paper-level detail.