CSCE 585: Machine Learning Systems

Lecture 3: How to Read an MLSys Paper?

Pooyan Jamshidi

UNIVERSITY OF South Carolina

Fall 2024 | Course Website: https://pooyanjamshidi.github.io/mls/

Objectives

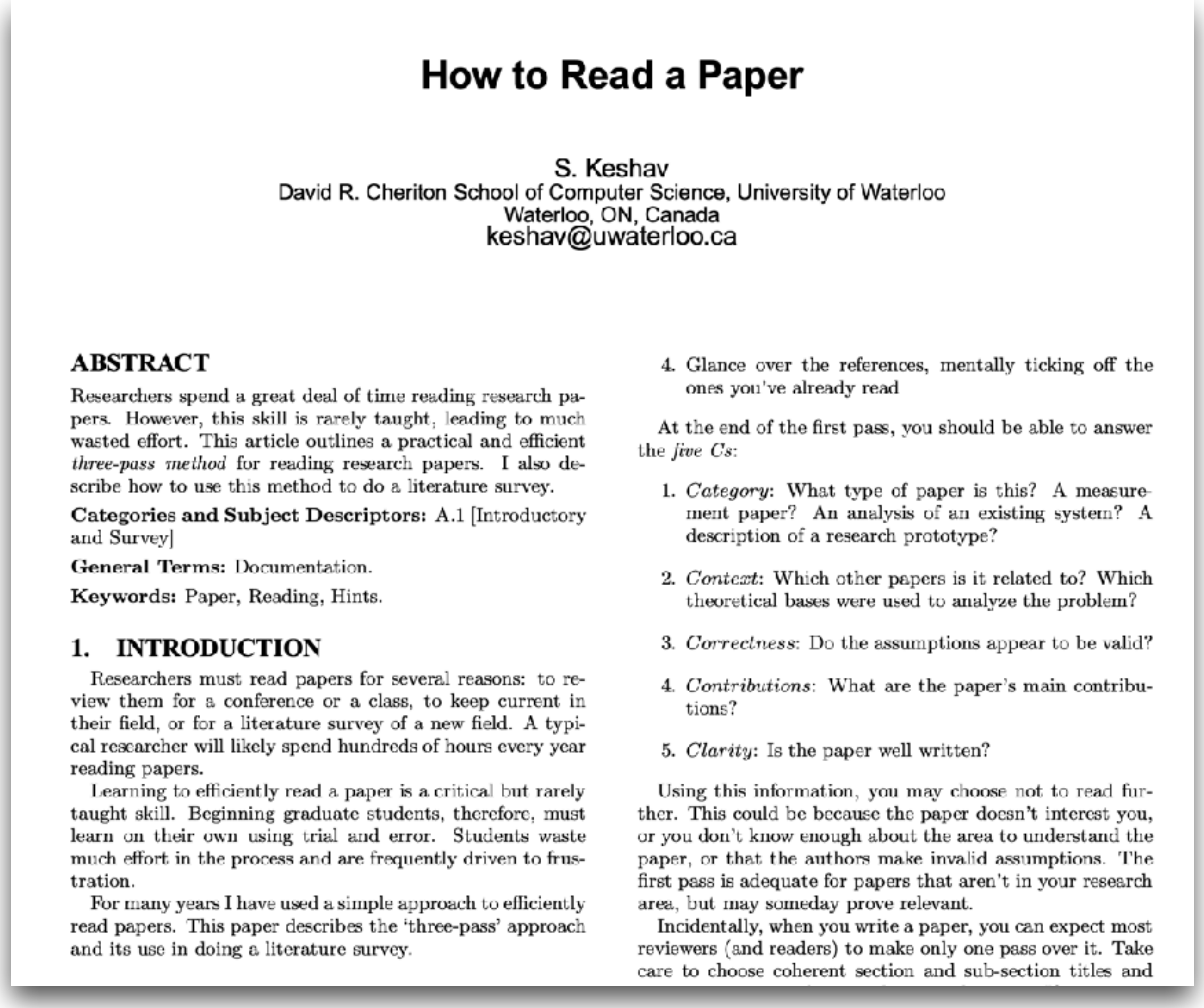
- Apply a **structured method** for reading MLSys papers.
  - The framework was adapted from a classic paper: S. Keshav's "How to Read a Paper."

- Efficiently extract **key ideas**, **contributions**, and practical **applications** relevant to machine learning systems.

- Break down papers to assess their **impact** on both Systems and ML.

Three-Pass Approach to Reading MLSys Papers

Adapted from Keshav



How to Read a Paper

S. Keshav
David R. Cheriton School of Computer Science, University of Waterloo
Waterloo, ON, Canada
keshav@uwaterloo.ca

ABSTRACT

Researchers spend a great deal of time reading research papers. However, this skill is rarely taught, leading to much wasted effort. This article outlines a practical and efficient *three-pass method* for reading research papers. I also describe how to use this method to do a literature survey.

Categories and Subject Descriptors: A.1 [Introductory and Survey]

General Terms: Documentation.

Keywords: Paper, Reading, Hints.

1. INTRODUCTION

Researchers must read papers for several reasons: to review them for a conference or a class, to keep current in their field, or for a literature survey of a new field. A typical researcher will likely spend hundreds of hours every year reading papers.

Learning to efficiently read a paper is a critical but rarely taught skill. Beginning graduate students, therefore, must learn on their own using trial and error. Students waste much effort in the process and are frequently driven to frustration.

For many years I have used a simple approach to efficiently read papers. This paper describes the 'three-pass' approach and its use in doing a literature survey.

4. Glance over the references, mentally ticking off the ones you've already read

At the end of the first pass, you should be able to answer the *five Cs*:

1. *Category*: What type of paper is this? A measurement paper? An analysis of an existing system? A description of a research prototype?
2. *Context*: Which other papers is it related to? Which theoretical bases were used to analyze the problem?
3. *Correctness*: Do the assumptions appear to be valid?
4. *Contributions*: What are the paper's main contributions?
5. *Clarity*: Is the paper well written?

Using this information, you may choose not to read further. This could be because the paper doesn't interest you, or you don't know enough about the area to understand the paper, or that the authors make invalid assumptions. The first pass is adequate for papers that aren't in your research area, but may someday prove relevant.

Incidentally, when you write a paper, you can expect most reviewers (and readers) to make only one pass over it. Take care to choose coherent section and sub-section titles and

First Pass: Get the Big Picture

The goal of the first pass is to gain a general understanding of the paper and decide whether it’s worth a deeper dive.

- **Look at the title, abstract, and introduction**:
  - What problem is the paper trying to solve? (Is it model- or system-centric?)
  - Why is this problem important in the context of ML systems?
- **Scan through headings, sections, and conclusions**:
  - What are the key contributions and insights?
  - Look for performance metrics, system architecture, or pipeline innovations.
- **Examine figures and tables**:
  - What are the benchmarks and key performance indicators?
  - Check resource use, throughput, or latency improvements.

Outcome: After the first pass, decide whether to proceed with a deeper analysis. At this stage, aim to understand **what problem is being solved and why it matters**.

Second Pass: Grasp the Paper's Content

In the second pass, read the paper more carefully to understand the method, results, and implications. Focus on the core technical contributions.

Read with focus:

- Carefully read the methodology and system design sections.
- For MLSys papers, pay attention to how system components are optimized or how inference is scaled (e.g., InferLine).

Follow the argument:

- Track how the authors move from identifying a problem to presenting a solution.
- For inference pipeline papers, analyze how system trade-offs (latency, resource scaling) are justified.

Focus on performance analysis:

- How do the proposed methods compare to existing systems?
- Are the results benchmarked against real-world systems? How practical are the improvements?

Outcome: By the end of the second pass, you should understand **the technical content**, including **the methodology, the system’s architecture**, and **the performance benchmarks**.

Third Pass: Dive into the Details

The third pass is for those deeply interested in the paper, such as students replicating the work or integrating it into a larger project. Here, focus on fine details, assumptions, and limitations.

- **Identify assumptions:**
  - ○ Are there assumptions about system hardware, network conditions, or data distribution?
  - ○ For example, in InferLine, are the inference optimizations hardware-specific?
- **Critique the methodology:**
  - ○ Can the experiments be reproduced in different environments (e.g., edge, cloud)?
  - ○ What are the limitations of the model or system (e.g., resource constraints, cost inefficiencies)?
- **Look for insights beyond the paper:**
  - ○ How could this work be extended? Could it be integrated into your ongoing project or system architecture?
  - ○ Do you identify any insightful differences between InferLine and IPA Saeid Ghafouri et al?

Outcome: After the third pass, you should be able to **reproduce results**, **suggest improvements**, and understand **how the system fits into larger architectures**.

Additional Tips

Use Tools for Reproducibility:

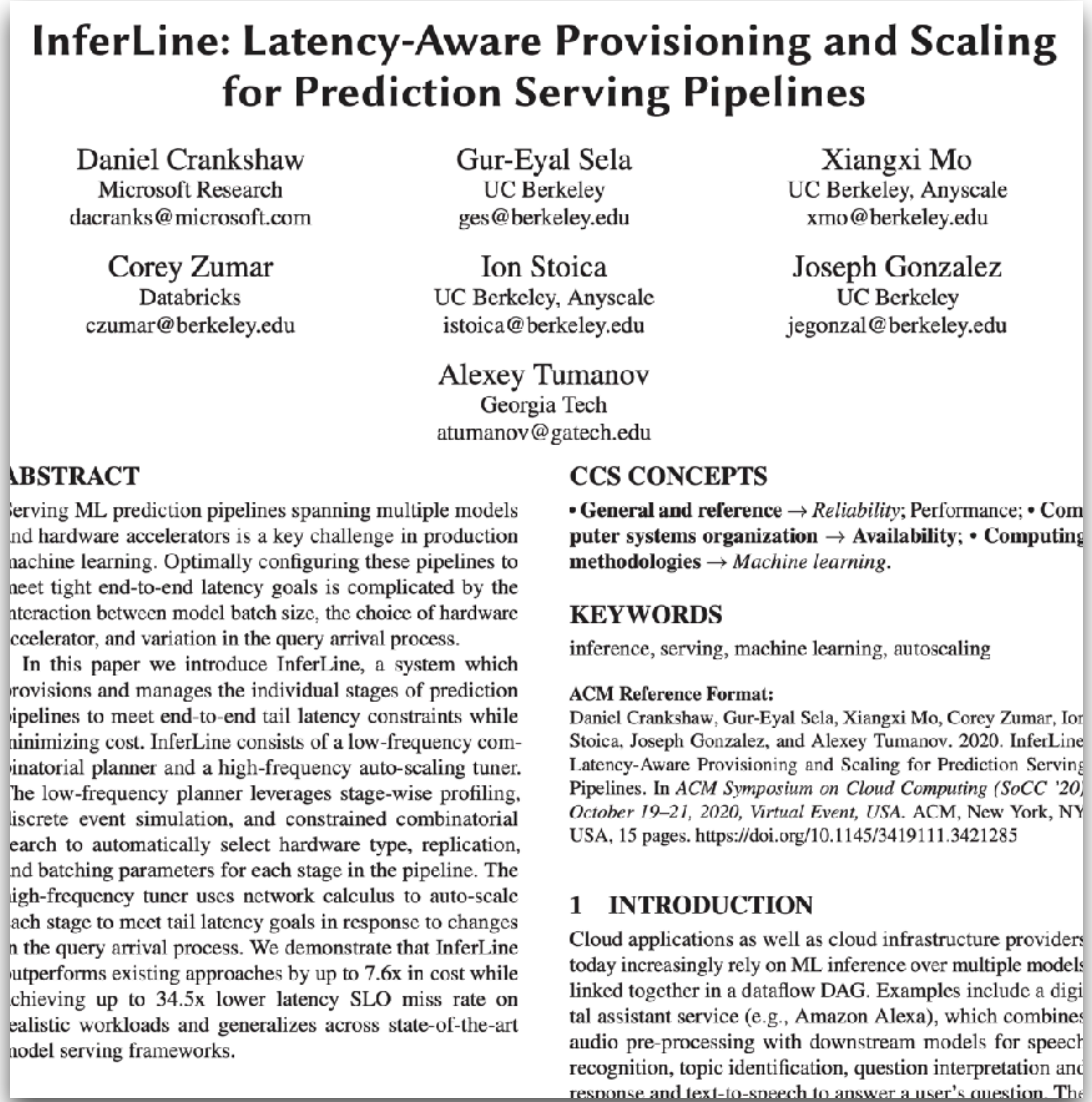
- Modern ML systems papers often include open-source code. So, that is why we are emphasizing replication of the results reported in the papers.
- GitHub, Docker, or cloud services (e.g., Chameleon).

Question the Author's Choices:

- Why did the authors choose this particular model architecture or hardware setup?
- Could another system (e.g., Clipper or Hydra) outperform the solution proposed in the paper?

Assignment 3.0

Read the paper following the 3-pass approach.



InferLine: Latency-Aware Provisioning and Scaling for Prediction Serving Pipelines

Daniel Crankshaw
Microsoft Research
dacranks@microsoft.com

Gur-Eyal Sela
UC Berkeley
ges@berkeley.edu

Xiangxi Mo
UC Berkeley, Anyscale
xmo@berkeley.edu

Corey Zumar
Databricks
czumar@berkeley.edu

Ion Stoica
UC Berkeley, Anyscale
istoica@berkeley.edu

Joseph Gonzalez
UC Berkeley
jegonzal@berkeley.edu

Alexey Tumanov
Georgia Tech
atumanov@gatech.edu

BSTRACT

erving ML prediction pipelines spanning multiple models
nd hardware accelerators is a key challenge in production
achine learning. Optimally configuring these pipelines to
eet tight end-to-end latency goals is complicated by the
nteraction between model batch size, the choice of hardware
ccelerator, and variation in the query arrival process.

In this paper we introduce InferLine, a system which
rovisions and manages the individual stages of prediction
ipelines to meet end-to-end tail latency constraints while
inimizing cost. InferLine consists of a low-frequency com-
inatorial planner and a high-frequency auto-scaling tuner.
he low-frequency planner leverages stage-wise profiling,
iscrete event simulation, and constrained combinatorial
earch to automatically select hardware type, replication,
nd batching parameters for each stage in the pipeline. The
igh-frequency tuner uses network calculus to auto-scale
ach stage to meet tail latency goals in response to changes
 the query arrival process. We demonstrate that InferLine
utperforms existing approaches by up to 7.6x in cost while
chieving up to 34.5x lower latency SLO miss rate on
ealistic workloads and generalizes across state-of-the-art
odel serving frameworks.

CCS CONCEPTS

• **General and reference** → *Reliability*; Performance; • **Com
puter systems organization** → **Availability**; • **Computing
methodologies** → *Machine learning*.

KEYWORDS

inference, serving, machine learning, autoscaling

ACM Reference Format:
Daniel Crankshaw, Gur-Eyal Sela, Xiangxi Mo, Corey Zumar, Ion
Stoica, Joseph Gonzalez, and Alexey Tumanov. 2020. InferLine
Latency-Aware Provisioning and Scaling for Prediction Serving
Pipelines. In *ACM Symposium on Cloud Computing (SoCC '20)
October 19–21, 2020, Virtual Event, USA*. ACM, New York, NY
USA, 15 pages. https://doi.org/10.1145/3419111.3421285

1 INTRODUCTION

Cloud applications as well as cloud infrastructure provider
today increasingly rely on ML inference over multiple model
linked together in a dataflow DAG. Examples include a digi
tal assistant service (e.g., Amazon Alexa), which combine
audio pre-processing with downstream models for speech
recognition, topic identification, question interpretation and
response and text-to-speech to answer a user's question. The

Applying the Three-Pass Approach to an MLSys Paper: InferLine Example

Assignment 3.1: Please write only one or at most two sentences to describe each item.

- **First Pass:**
  - **What:**
  - **Why**:
- **Second Pass:**
  - **Methodology**:
  - **Performance**:
- **Third Pass:**
  - **Critique**:
  - **Extension**:

Applying the Three-Pass Approach

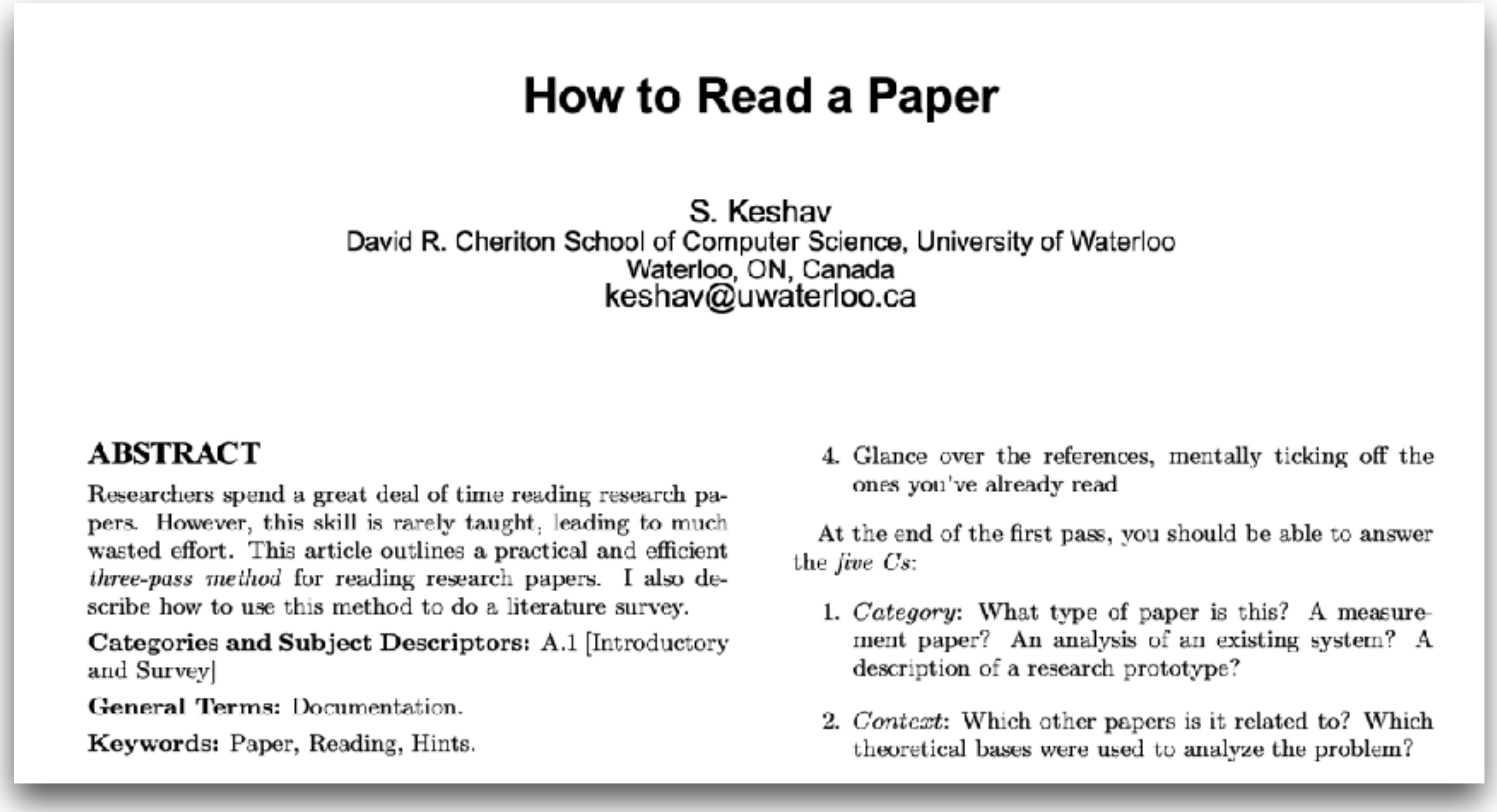
Assignment 3.2

- **Task**: Read “InferLine” and use the three-pass method to:

1. Summarize the paper in 200 words after the first pass.
2. Write a 1-page critique after the second pass, discussing the methodology and performance results.
3. For extra credit, suggest improvements or extensions to the system after completing the third pass.

Conclusion

- The three-pass method provides **a systematic approach** to understanding, both high-level ideas and technical details in MLSys papers.
- I strongly encourage you to apply this technique to efficiently **digest complex research** while maintaining focus on key innovations in machine learning systems.
- I aim to update these slides based on the discussions regarding papers that we will have throughout the semester. So, I would love to hear your thoughts regarding anything that **particularly worked or did not work for you** specifically when you did the reading exercises.



How to Read a Paper

S. Keshav
David R. Cheriton School of Computer Science, University of Waterloo
Waterloo, ON, Canada
keshav@uwaterloo.ca

ABSTRACT

Researchers spend a great deal of time reading research papers. However, this skill is rarely taught, leading to much wasted effort. This article outlines a practical and efficient *three-pass method* for reading research papers. I also describe how to use this method to do a literature survey.

Categories and Subject Descriptors: A.1 [Introductory and Survey]

General Terms: Documentation.

Keywords: Paper, Reading, Hints.

4. Glance over the references, mentally ticking off the ones you've already read

At the end of the first pass, you should be able to answer the *five Cs*:

1. *Category*: What type of paper is this? A measurement paper? An analysis of an existing system? A description of a research prototype?
2. *Context*: Which other papers is it related to? Which theoretical bases were used to analyze the problem?

How to Read an MLSys Paper?