Requirements are not contractual constraints, or user needs. They are social infrastructure that reflect, reinforce, and amplify the matrix of oppression.
Whether it's youth debugging an animation, a developer auditing a classifier for bias, or a teacher linking code to oppression, I study our individual and collective struggle to understand computing and harness it for play, power, equity, and justice.

I work with outstanding students and vibrant communities on this work, publishing it in Computing Education, Human-Computer Interaction, and Software Engineering communities, and then blogging, presenting, teaching, synthesizing, and applying it in the world.

I have expertise in debugging, program understanding, explainability, end-user programming, human aspects of software engineering, broadening participation in computing, assessment, and HCI education. I also have broad methodological expertise, including the design and implementation of user interfaces, programming languages, and developer tools; HCI methods; multivariate null hypothesis significance testing; Bayesian modeling; qualitative methods such as interviews, surveys, content analysis, and fieldwork; data science methods; and argumentation.
Lecture 3 Administrivia

A1 & first weekly log grades released

The purpose of a research log is so you (and course staff) can track and record progress! Make sure to format it with each date and hours spent, like so:

A2 (Related Work) is due next Thursday

Afterwards, for A3, you will write an introduction section to your proposal.

Your main goal (outside of writing) is to make progress on research! We will continue to have weekly check-ins.

Week 3: 4/16 - 4/22

Goals:
1. Debrief with Mohsen about sub-permutations implementation
2. Check orthogonality of feature vectors (determine if method doesn’t work or if our implementation is wrong)
3. Come up with our own method to make feature vectors orthogonal

Accomplished:
Monday 4/16: (0.5hrs)
- Met with Mohsen. We let Mohsen know about our success with the permutation encoding.
How do we get to the point where we know what has been done, and why our idea is different, new, and exciting?

Bit flip: articulating an assumption present in all prior work that you are breaking

Literature search process:

Iterative expansion of the most relevant work from the set of papers you’ve seen so far
How do we articulate our project persuasively to a peer? A bit flip isn’t enough on its own.

If we can’t explain the project clearly enough for another researcher in the same area to understand it, we don’t really understand our project ourselves.

(This happens more often than you might think…it’s hard!)
INTRODUCTION
Crowdsourcing mobilizes a massive online workforce into collectives of unprecedented scale. The dominant approach for crowdsourcing is the microtask workflow, which enables contributions at scale by modularizing and pre-specifying all actions [7, 55]. By drawing together experts [70] or amateurs [6], microtask workflows have produced remarkable success in robotic control [48], data clustering [12], galaxy labeling [54], and other goals that can be similarly pre-specified. However, goals that are open-ended and complex, for example invention, production, and engineering [42], remain largely out of reach. Open-ended and complex goals are not easily adapted to microtask workflows because it is difficult to articulate, modularize, and pre-specify all possible actions needed to achieve them [71, 80]. If crowdsourcing remains confined to only the goals so predictable that they can be entirely pre-defined using workflows, crowdsourcing’s long-term applicability, scope and value will be severely limited.

In this paper, we explore an alternative crowdsourcing approach that can achieve far more open-ended and complex goals: crowds structured like organizations. We take inspiration from modern organizations because they regularly orchestrate large groups in pursuit of complex and open-ended goals, whether short-term like disaster response or long-term like spaceflight [8, 9, 63]. Organizations achieve this complexity through a set of formal structures — roles, teams, and hierarchies — that encode responsibilities, interdependencies and information flow without necessarily pre-specifying all actions [15, 83].

We combine organizational structures with computational crowdsourcing techniques to create flash organizations, rapidly assembled and reconfigurable organizations composed of online crowd workers (Figure 1). We instantiated this approach in a crowdsourcing platform that computationally convenes large groups of expert crowd workers and directs their efforts to achieve complex goals such as product design, software development and game production.

We introduce two technical contributions that address the central challenges in structuring crowds like organizations. The first problem: organizations typically assume asset specificity, the ability for organization members to develop effective collaboration patterns by working together over time [83]. Clearly, crowds, with workers rapidly assembled on-demand from platforms such as Upwork (www.upwork.com), do not offer asset specificity. So, our system encodes the division of labor into a de-individualized role hierarchy, inspired by movie crews [2] and disaster response teams [8], enabling workers to coordinate using their knowledge of the roles rather than their knowledge of each other.

The second problem: organizational structures need to be continuously reconfigured so that the organization can adapt as work progresses, for example by changing roles or adding teams [9, 63, 83]. Coordinating many workers’ reconfigurations in parallel, however, can be challenging. So, our system enables reconfiguration through a model inspired by version control: workers replicate (branch) the current organizational structure and then propose changes (pull requests) for those up the hierarchy chain to review, including the addition of new tasks or roles, changes to task requirements, and revisions of the organizational hierarchy itself. Enabling new forms of organization could have dramatic impact: organizations have become so influential as the backbone of modern economies that Weber argued them to be the most important social phenomenon of the twentieth century [82]. Flash organizations advance a future where organizations are no longer anchored in traditional Industrial Revolution-era labor models, but are instead fluidly assembled and re-assembled from globally networked labor markets. These properties could eventually enable organizations to adapt at greater speed than today and prototype new ideas far more quickly.

In the rest of the paper, we survey the foundations for this work and describe flash organizations and their system infrastructure. Following this review, we present an evaluation of three flash organizations and demonstrate that our system allows crowds, for the first time, to work iteratively and adaptively to achieve complex and open-ended goals. The three organizations used our system to engage in complex collective behaviors such as spinning up new teams quickly when unplanned changes arose, training experts on-demand in areas such as medical privacy policy when the crowd marketplace could not provide the expertise, and enabling workers to suggest bottom-up changes to the work and the organization.?
How many papers have you read?
Agenda

Time management tips
What is an introduction?
How to write an introduction
Time management tips
How do I reduce anxiety about assignments?

Sometimes, assignments seem daunting because they're so big. (For example, conducting a literature search with 15 papers.)

For research, this is even more the case — a CS research project takes months, even years, to complete.

**Scenario one (common):**

😄 I have plenty of time!

😱 I have so much work to do!

April 2022

Sept 2022 (deadline)
Tip 1: Break down to “unit” tasks

Each research paper, or assignment, can be broken down even further.

We’re scaffolding your final proposal now (e.g., into a related works assignment, an intro assignment...)

But even these assignments can be broken down more.

The key is to stop when (1) you can define the task in such clarity that you can sit down and do it without thinking about what to do, and (2) accomplish it in a single sitting.
Tip 1: Break down to “unit” tasks

Example, for assignment 2:

1. Read 15 papers
   - Read 5 papers from nearest neighbor paper
   - Read 2 more papers in topic area X, branching off existing paper
   - Read 3 more papers in topic area Y
   - Read 5 more papers in topic area Z

2. Affinity map papers
   - Transfer papers to post-it notes, group together
   - Write thesis statements for each group

3. Write related work section
   - Set up Overleaf, import citations
   - Outline section with thesis statements
   - Figure out bit flip for each paragraph
   - Draft paragraphs and put in citations
   - Edit section
Tip 2: Time estimate your unit tasks

- Read 5 papers from nearest neighbor paper • 2 hours
- Read 2 more papers in topic area X, branching off existing paper • 30 min
- Read 3 more papers in topic area Y • 30 min
- Read 5 more papers in topic area Z • 1 hour

- Transfer papers to post-it notes, group together • 30 min
- Write thesis statements for each group • 30 min

- Set up Overleaf, import citations • 10 min
- Outline section with thesis statements • 20 min
- Figure out bit flip for each paragraph • 30 min
- Draft paragraphs and put in citations • 1.5 hours
- Edit section • 30 min

4 hours reading
1 hour affinity mapping
4 hours writing
Tip 2: ...then multiply by 2.5

10 hours reading

"This seems like a lot! I thought this was supposed to make me feel better, but now the high numbers just give me more anxiety."

2.5 hours affinity mapping

Studies show that people starting out in their careers always drastically underestimate how long work will take, but you’ll get better as you do it more.

8 hours writing

It’s better to overestimate and then have time to relax!

Rockquemore and Laszlofy (2008)
Tip 3: Schedule it in your calendar & time track

Now that you know how long things will take, create a meeting with yourself to do it. Show up for yourself.

Note how much time it actually took.

Afterwards, go back, reflect, and add up your hours. Feel proud of the work you put in!

Tip 4: Be strategic about your energy

**Eisenhower Matrix**

<table>
<thead>
<tr>
<th>Important</th>
<th>Urgent</th>
<th>Nonurgent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class assignments</td>
<td>Research tasks on a long project</td>
</tr>
<tr>
<td></td>
<td>the day before</td>
<td></td>
</tr>
<tr>
<td></td>
<td>they're due</td>
<td></td>
</tr>
<tr>
<td>Nonimportant</td>
<td>Emails</td>
<td>Social media</td>
</tr>
</tbody>
</table>

Try to do your nonurgent but important tasks first thing in your day (or whenever you have the most energy) as much as possible

“Eisenhower Matrix”
Questions?
Reactions?
Your own tips?
Architecture of an Introduction
What is an Introduction?

The Introduction makes the case for your research, in brief.

Jennifer Widom, Dean of Engineering:

“The Introduction is crucially important. By the time a referee has finished the Introduction, they've probably made an initial decision about whether to accept or reject the paper — they'll read the rest of the paper looking for evidence to support their decision.

A casual reader will continue on if the Introduction captivated them, and will set the paper aside otherwise. Again, the Introduction is crucially important.”

https://cs.stanford.edu/people/widom/paper-writing.html#intro
Think of it this way...

By this point, the video has hopefully made clear to you what it’s about, and you’ve made a decision about whether to watch the rest of it.
Each introduction makes the case for two things:

1) The **problem**: why do we care about the problem you’re solving?

2) The **solution**: why is your approach creative and correct?
...great, Jingyi, thanks. But how do we actually do this?
The Problem

By yourself, jot down an explanation of the problem that your project is working on.

How clearly do you understand your own problem?

How clearly do you understand your own bit flip?

Either describe the problem of the overall project, or of your specific contribution.

(2 minutes)
Unpacking the problem

The Introduction’s goal isn’t just to set up the problem, it’s to convey the solution as well. To do that effectively, your problem statement needs to set up the **bit flip**.

For this to succeed, **the bit needs to integrated as part of the problem statement.**
Problem motivation

Explain the main problem that you’re trying to solve:

- Networks are hard to (re)configure
- Interactions with computers are stuck on flat glass displays
- Generative AI models are challenging to evaluate

Use citations to back up your claims about the existence of the problem, and why we should care about solving it.
Set up the bit

Answer the question, "Why isn't this problem solved yet?" by setting up the bit that you're going to flip:

- Networks are configured in hardware
- To break out of glass screens, outputs have been designed into the physical world.
- Generative model evaluations have been automated, but these are proxies at best.

This is a summary of related work that is in service of your bit set up.
Funnel

A good introduction is like a funnel. Your first sentence should be very broad to introduce the domain of your project, and each sentence narrows it down to your exact bit.

Problem motivation
Set up the bit
Solution (bit flip)
First sentence = OK, we are in the domain of 2D illustrated characters, and we understand why this domain is important (i.e., it’s a widespread type of media we encounter).
1 INTRODUCTION
Illustrated 2D characters are prevalent across many visual applications, including storytelling, advertising, animation, and games. Recently, “mix-and-match” 2D character creation tools have emerged that make character design more accessible to a broader range of users.

Second sentence = Introducing the bit. Specifically, we are working with mix & match 2D character creation tools in this project.
1 INTRODUCTION

Illustrated 2D characters are prevalent across many visual applications, including storytelling, advertising, animation, and games. Recently, “mix-and-match” 2D character creation tools have emerged that make character design more accessible to a broader range of users. Instead of starting from scratch, tools and templates like Open Peeps [24], Blush [6], Character Creator [11], and Picrew [33] allow users to assemble characters by choosing from discrete sets of predefined variations for attributes such as body shape, skin color, and clothing. These variations are represented as 2D artwork layers that are composited together to produce the final character appearance. We refer to the artwork that represents variations of the character’s unadorned body—such as different shapes, poses, and skin colors—as body layers. Likewise, we define accessory layers as the various objects (e.g., clothing, hairstyles, facial features, accoutrements, etc.) that adorn the base layers. With mix-and-match tools, users can quickly explore a range of character designs by selecting different combinations of body and accessory layers.

Bit = mix & match 2D character creation tools

The rest of the paragraph is dedicated to giving examples and explaining how these tools work.

The first paragraph usually offers important background knowledge for the reader on the “state of the world.”

Problem motivation.

(Note: Your Introduction cites essential work for setting up the bit. Exhaustive lit review goes in your Related Work.)
Now, the second paragraph actually introduces the problem, usually as the topic sentence. The rest of the paragraph expands on this problem in more detail and sets up the bit.
Try again: The Problem

Get in a group of 3.

Explain the problem you’re working on. Practice funneling. Just the bit, not the flip yet. (1 min each)

How clearly do you understand your partners’ problems?

Problem motivation
Set up the bit
Solution (bit flip)
Architecture of an intro

Problem motivation
Set up the bit
Solution (bit flip)
The Solution

By yourself, jot down the approach your project is taking.

How clearly do you understand your own bit flip?

How clearly do you understand how exactly the project is going to instantiate that bit flip in a specific system, algorithm, or design?

Either describe the solution of the overall project, or of your specific contribution.

(2 minutes)
Unpacking the solution

The solution has to explain two things: what the big idea is, and how that big idea gets instantiated in the specific context of this problem.

(Even if someone hears your bit flip that you want to introduce recurrence inside the neural network, they may still have no idea how that actually connects to the problem of language generation.)
Flip the bit

The topic sentence of this paragraph is the thesis statement of your entire research project.

Pivot off of the bit you set up to flip the bit. Explain why flipping the bit is a good idea for the problem at hand.

It should now be obvious to a reader given the prior paragraph that this research is novel, since you have proven that nobody else has flipped that bit.
original character designers may want to modify a character’s body shape or pose away from the predefined options. For example, for the character shown in Fig 1, a user may want to explore different torso proportions to better match the character to their own body, or animate an “idle” squatting animation when making the character playable in a video game. The problem is that current mix-and-match tools do not support such modifications. Since the character’s accessory layers are only designed to fit with the predefined body layers, edits to the body shape or pose produce artifacts in the accessory layers. For example, in Fig 1D, we see the new body shape protrude out of the clothing—since the independent accessory layers are not constrained to the body layers in any way, edits to the body do not propagate to them. To fix these artifacts, users would have to manually edit or redraw all the accessory layers to fit the edited body.

The goal of our work is to make mix-and-match character creation tools more flexible by allowing users to modify a character’s body layers while automatically adapting the accessory layers to fit.
At this point, the reader understands the idea that you're proposing, but it's still very high level. In this paragraph, map that idea onto a concrete instantiation.

Typically, this is where the system or algorithm that you’re creating gets a name. Explain its architecture or design at a high level. Make clear how this architecture or design is an instance of the bit flip.
The goal of our work is to make mix-and-match character creation tools more flexible by allowing users to modify a character’s body layers while automatically adapting the accessory layers to fit. Our approach is to automatically generate a rig, a set of constraints that specify how changes in the body layers should transfer to the accessory layers, for each accessory layer. Each rig captures and preserves important relationships between artwork layers, such as where and how they should remain attached with respect to each other under deformation. Specifically, we introduce four types of constraints that represent common layer-to-layer interactions: (1) occlusions (e.g., arm against sleeve), (2) attachment at a point (e.g., brooch on sweater), (3) coincident boundaries (e.g., lapel on jacket), (4) overlapping regions (e.g., sleeve to bodice near the armpit). Our rigs model these interactions via inter-layer spatial constraints that cause accessory layers to deform as the body layers change. This approach allows users to customize a character’s body shape and pose while still making use of all the accessory artwork that was designed for the default body shapes and poses. Fig 1F–H shows different accessories adapting to the same body deformation specified in Fig 1C.

\[
\text{instantiation} = \text{constraints in the form of rigs, and an algorithm to automatically generate these rigs}
\]

The rest of the paragraph details how the rigs work.
Try again: The Solution

In your group, explain the approach your project is taking. (1 min each).

How clearly do you understand your partner’s bit flip?

How clearly do you understand how exactly the project is going to instantiate that bit flip in a piece of software?
Evaluation

How did you prove that your bit flip is successful at solving the problem?

We obviously haven’t covered evaluation yet in this course, so for now you’ll need to take your best guess.

How would you convince a critical reader that flipping the bit solved the problem better than the prior work?
Our main contribution is the definition of the constraints and automated techniques for rigging accessories with these constraints such that accessory layers appropriately adapt to deformations on the body layers. We use our approach to automatically rig a wide range of character accessories derived from existing mix-and-match data sets and show the results for various modifications to the shape and pose of the character. In addition, we demonstrate two applications where our rigs enable continuous edits that propagate to all the accessories: an interactive character customization interface where users modify various body shape parameters, and animations that change the body shape and pose over time.

In the last paragraph, often the contribution is explicitly stated.

evaluation = creating a large example space of images and applications to prove these rigs work across diverse artwork
Implications

If you’re right and the bit flip is how everyone should be approaching this problem from now on, what implications are there for the field?

This is your chance to stand on a small soapbox:

Will it change the contexts in which we use this technology? Will it broaden usage?

But don’t overplay your hand:

It probably won’t change all of computing.
In brief: use your literature search to motivate your problem and set up a bit.

Then, flip the bit and argue persuasively that this will address the problem. Explain how this solution gets built into your system or model.
Genres of Research
First, find your genre

Recall:

Activity tracking hardware (Method)
- Specialized
  - Exercise
  - Nearest neighbor
- Commodity
  - Your project
  - Not here!

Activity being tracked (Problem)
- Diet
- Exercise

As mentioned last lecture, there are a few different kinds of paper that are common:

- New problem / old method
- Old problem / new method

Exercise

Diet

Nearest neighbor

Your project

Not here!
State of the literature

Answer a new question with an old method

Social media disclosures of mental illness

Answer an old question with a new method

Tie strength and Facebook use

Solve a new problem with a new technique

Hard to convince the world
Why only make one move?

When making an argument, you want to introduce one major new idea, to minimize the new ideas your listener needs to absorb.

Certain ideas already have warrants in the literature: prior work already has proven their legitimacy. A warrant is a free pass!

Old problem: the problem already has a warrant in the literature.

Visual question answering is a legitimate task; mission critical code should be proven correct; interaction should not happen on panes of glass

Old solution: the solution already has a warrant in the literature.

Sensor fusion into features for an ML system; transformer architectures for NLP; tangible interaction; self-play in reinforcement learning
Why only make one move?

Typically you are spending the introduction making the case for your new idea. If you are trying to make the case for both a new problem and a new solution, a reader might disagree with either.

This is not to say that you can’t do new problem / new solution; just that it’s a riskier, varsity maneuver.
How to Write
The Introduction
From genre to intro

Old problem / new method:

Motivate the problem via **prior work**, which has already established the problem

Set up the bit of **how all prior work tried to solve it**

Flip the bit — your new solution

Instantiate that new solution

Implications

New problem / old method:

Motivate the problem via **rhetoric**, drawing on prior work making supporting claims

Set up the bit: **prior work is not equipped for this problem**

Flip the bit — your new solution

Instantiate that new solution

Implications
Start with an outline

Problem motivation
Set up the bit
Flip the bit
Instantiate the bit flip
Evaluation
Implications

Your idea should be fully understandable with only six sentences, which can become your paragraph topic sentences.
Keep it taut!

Your goal is then to treat each outline point as a thesis sentence for the paragraph, and use the paragraph to prove that thesis. Don’t stray and make other interesting but un-useful points.

Avoid “tunneling” the reader to your point.

Instead, “zoom in” the reader to an uncharted island on a known map.
Your To-dos

1. Assignment 2 (due next Thurs)
   - Use the time management tips if you’re feeling overwhelmed! One step at a time.
   - Reminder: my OH are Weds 3-4pm on Zoom

2. Continue to update weekly logs

Exit ticket: http://tiny.cc/cs197-week3
Computer Science Research

Slide content shareable under a Creative Commons Attribution-NonCommercial 4.0 International License.