Arguing a Research Project

CS 197 | Stanford University | Arpita Singhal
cs197.stanford.edu

(Adapted from Prof. Michael Bernstein's slides)

Administrivia

You all have projects and groups at this point, and are digging in on those projects. Let us know if that's not the case.

Assignment 3 — Project Introduction — is out and due next week.

> Your main goal is to make self-guided progress on the project through the rest of the quarter! Assignments are parallel with project progress.
>
> Project and Assignments both happen in parallel. After A3, you will have only progress reports as your deliverables each week for a while.

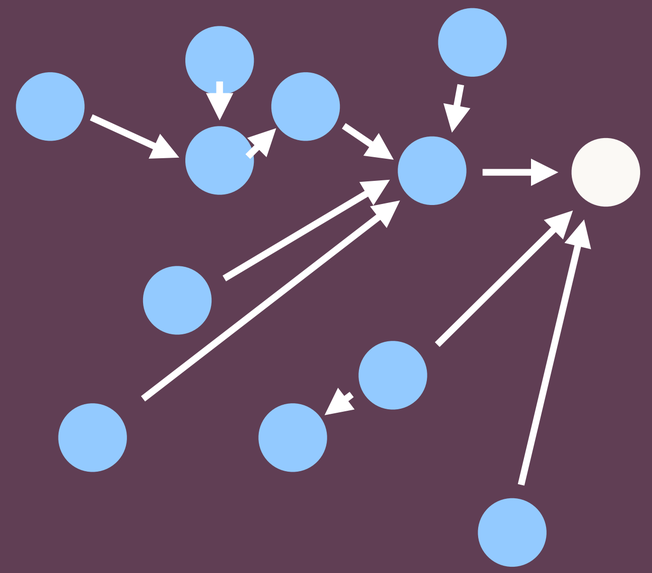
Last time

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



Today: how to explain your research and impact

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!)

Your goal

To convey your research to another person, you need to be able to explain what you're doing and why it matters.

As it turns out, this is exactly the same task as the Introduction to your paper. →

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

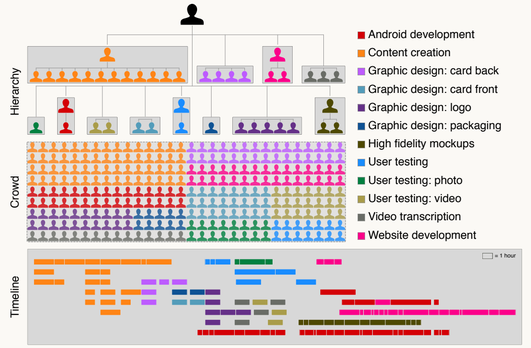


Figure 1: Flash organizations are crowds that are computationally structured like organizations. They enable automated hiring of expert crowd workers into role structures and continuous reconfiguration of those structures to direct the crowd's activities toward complex goals.

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.

Explaining a project

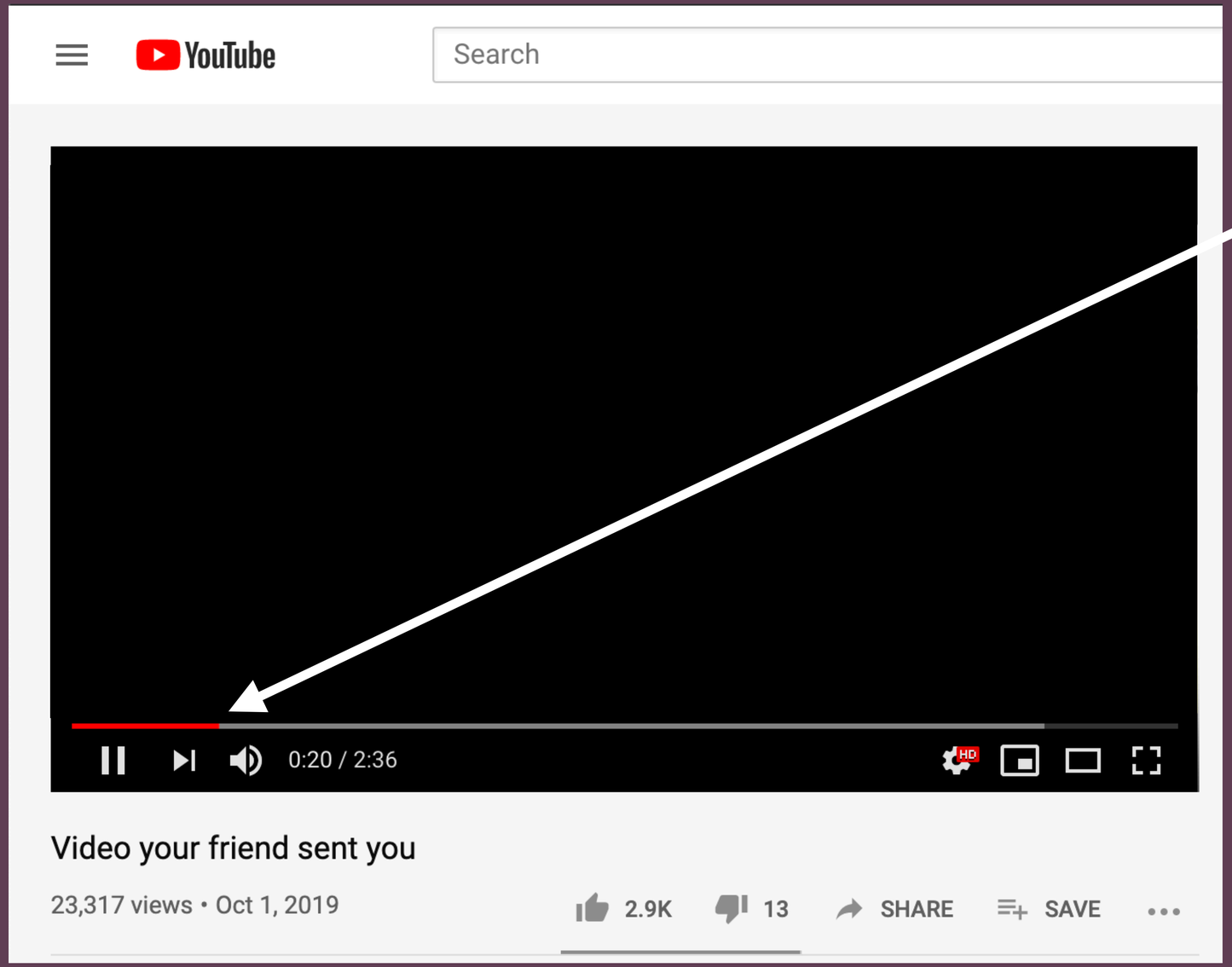
What is an Introduction?

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

Jennifer Widom:

> *"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.

Try it

Turn to a partner and explain your project to them [1 min each]

How clearly do you understand the problem that your partner is addressing?

How clearly do you understand the solution that your partner is proposing?

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?

Architecture of an intro

Problem

Solution

…great, Arpita, thanks. But how do we actually do this?

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
Problem
Set up the bit
Solution
Solution (bit flip)

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.

Problem motivation

Set up the bit

Solution (bit flip)

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: one that's directly in service of your bit.

Problem motivation

Set up the bit

Solution (bit flip)

Crowdsourcing platforms such as Amazon Mechanical Turk decentralize their workforce, designing for distributed, independent work [16, 42]. Decentralization aims to encourage accuracy through independent judgement [59]. However, by making communication and coordination more difficult, decentralization disempowers workers and forces worker collectives off-platform [41, 64, 16]. The result is disenfranchisement [22, 55] and an unfavorable workplace environment [41, 42]. Worse, while decentralization is motivated by a desire for high-quality work, it paradoxically undercuts behaviors and institutions that are critical to high-quality work. In many traditional organizations, for example, centralized worker coordination is a keystone to behaviors that improve work quality, including skill development [2], knowledge management [35], and performance ratings [58].

bit = decentralization

The rest of this introduction paragraph is dedicated to surveying related work with respect to how decentralization is architected, and to its outcomes.

Architecture of an intro

Problem statement

Set up the bit

Solution (bit flip)

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.)

Problem motivation
Problem motivation
Set up the bit
Set up the bit
Flip the bit
Solution (bit flip)
Instantiate the bit flip

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.

Problem motivation

Set up the bit

Flip the bit

Instantiate the bit flip

To address this reputation challenge, and with an eye toward other challenges that arise from decentralization, we draw inspiration from a historical labor strategy for coordinating a decentralized workforce: *guilds*. Worker guilds arose in the early Middle Ages, when workers in a trade such as silk were distributed across a large region, as bounded sets of laborers who shared an affiliation. These guilds played many roles, including training apprentices [18, 44], setting prices [45], and providing mechanisms for collective action [52, 49]. Especially relevant to the current challenge, guilds measured and certified their own members' quality [18]. While guilds eventually lost influence due to exerting overly tight controls on trade [45] and exogenous technical innovations in production, their intellectual successors persist today as professional organizations such as in engineering, acting and medicine [46, 33]. Malone first promoted a vision of online "e-lancer" guilds twenty years ago [40], but to date no concrete instantiations exist for a modern, online crowd work economy.

flip = re-centralizatize via guilds

The rest of the paragraph explains the high level idea.

Instantiate the bit flip

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.

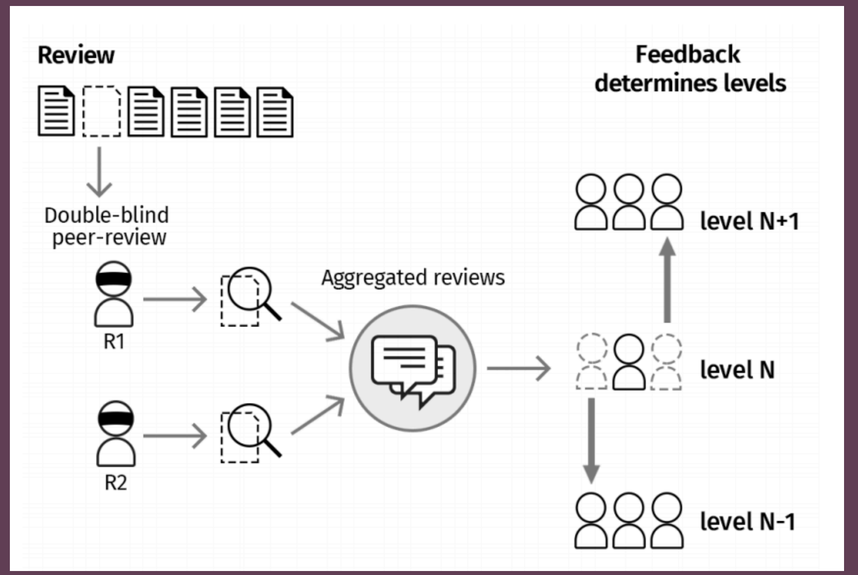
Problem motivation

Set up the bit

Flip the bit

Instantiate the bit flip

We present *crowd guilds*: crowd worker collectives that coordinate to certify their own members and perform internal feedback to train members (Figure 1). Our infrastructure for crowd guilds enables workers to engage in continuous double-blind peer assessment [30] of a random sample of members' task submissions on the crowdsourcing platform, rating the quality of the submission and providing critiques for further improvement. These peer assessments are used to derive guild levels (e.g., Level 1, Level 2) to serve as reputation (qualification) signals on the crowdsourcing platform. As workers gather positive assessments from more senior guild members, they rise in levels within the guild. Guilds translate these levels into higher wages by recommending pay rates for each level when tasks are posted to the platform. While crowd guilds focus here on worker reputation, our experiment implementation also explores how crowd guilds could address other challenges such as collective action (e.g., collectively rejecting tasks that pay too little), formal mentorship (e.g., repeated feedback and training), and social support (e.g., on the forums). Because



instantiation = crowd guilds system

The rest of the paragraph details how crowd guilds work.

Try again

Turn to your partner and explain your project. Cheat sheet to the right.

[1 min each]

How clearly do you understand your partner's problem?

How clearly do you understand your partner's solution?

Problem motivation

Set up the bit

Flip the bit

Instantiate the bit flip

Evaluation and Implications

We'll talk about evaluation later in the course.

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?

Problem motivation

Set up the bit

Flip the bit

Instantiate the bit flip

Evaluation

Implications

Architecture of an intro

Problem motivation

Set up the bit

Flip the bit

Instantiate the bit flip

Evaluation

Implications

So in brief: use your literature search to motivate your problem and set up a bit.

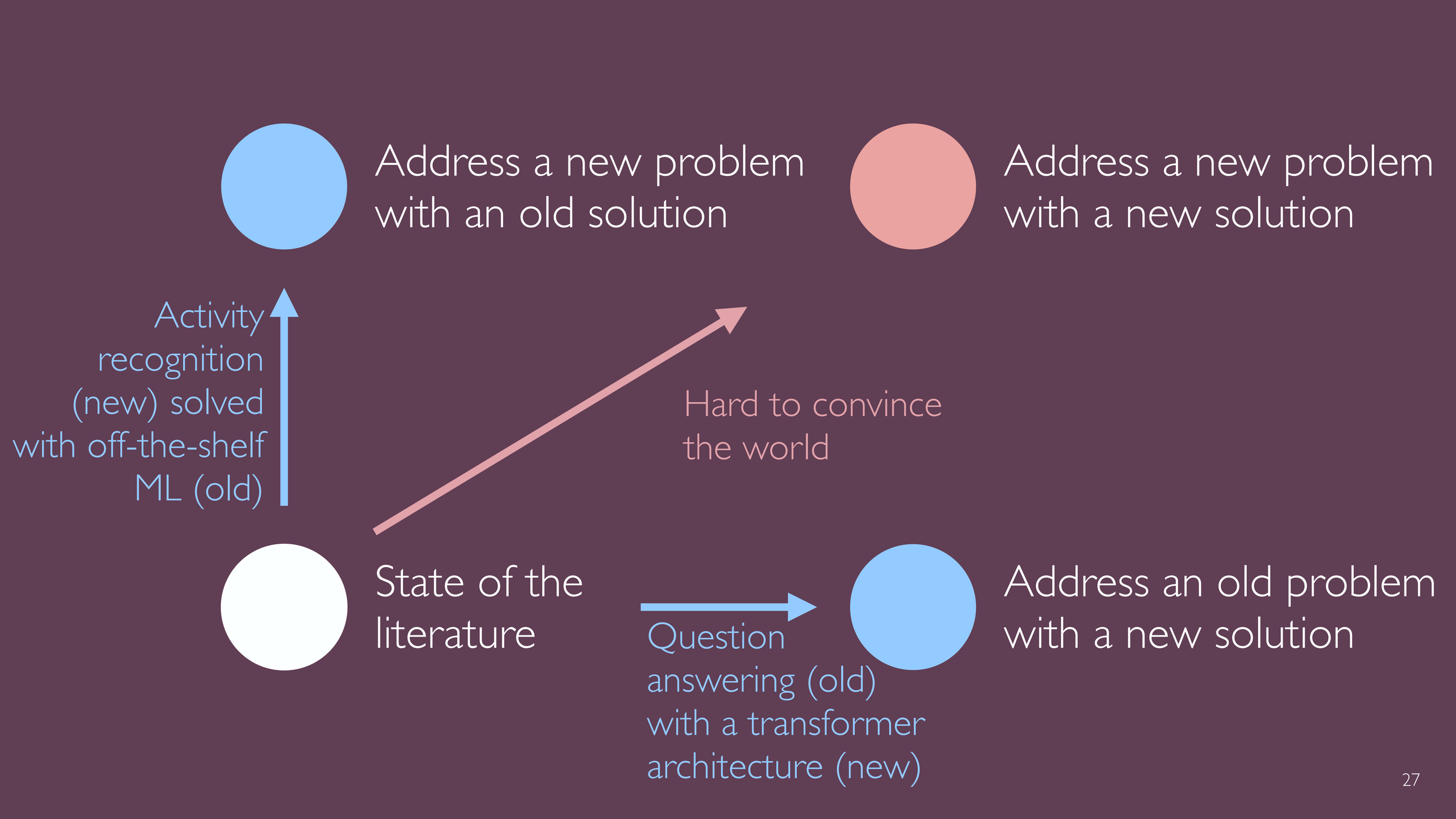
Then, flip the bit and argue persuasively that this will address the problem. Explain how this solution gets built into your system or model.

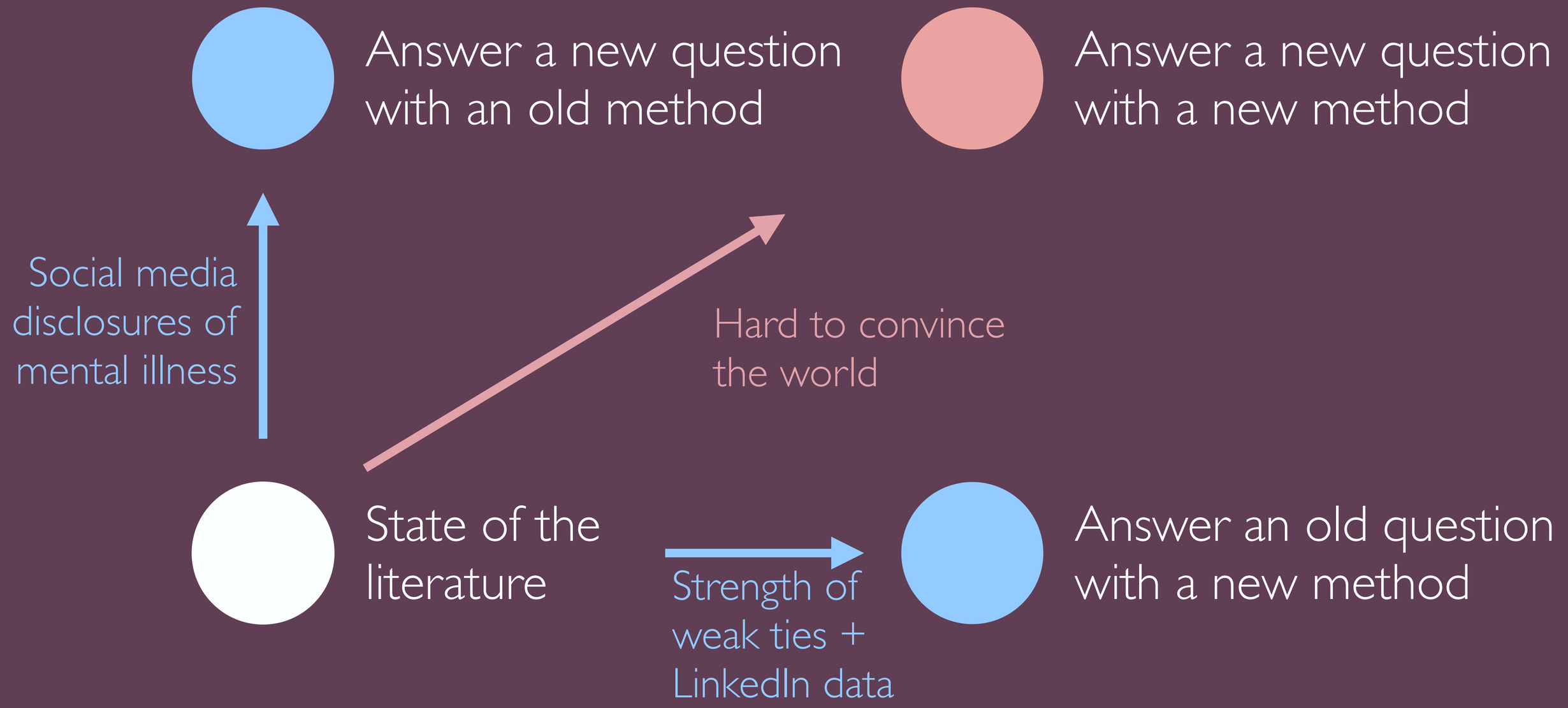
How to Write The Introduction

First, find your genre

There are a few different kinds of paper that are common:

- New problem / old solution
- Old problem / new solution


Address a new problem with an old solution
Address a new problem with a new solution
Activity recognition (new) solved with off-the-shelf ML (old)
Hard to convince the world
State of the literature
Question answering (old) with a transformer architecture (new)
Address an old problem with a new solution


Answer a new question with an old method
Answer a new question with a new method
Social media disclosures of mental illness
Hard to convince the world
State of the literature
Strength of weak ties + LinkedIn data
Answer an old question with a new method

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. A research paper typically only flips **one** bit.

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 risky varsity maneuver.

Use existing warrants

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

From genre to intro

Old problem / new solution:

- 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 solution:

- 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 — how your approach draws on known ideas
- Instantiate that solution
- Implications

Start with an outline

Your idea should be fully understandable with only six sentences, a topic sentence per paragraph:

Problem motivation

Set up the bit

Flip the bit

Instantiate the bit flip

Evaluation

Implications

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.

Assignment 3

Your group writes an Introduction to a paper for your project

- Outline the introduction
- Turn the outline into text
- 700-900 words

Details at cs197.stanford.edu

Computer Science Research