Vectoring in Research

CS 197 & 197C | Stanford University | **Sean Liu** & Lauren Gillespie
cs197.stanford.edu | cs197c.stanford.edu

Slides adapted from previous iterations of the course by Michael Bernstein and Jingyi Li

Vectoring in Research

CS 197 & 197C | Stanford University | **Sean Liu** & Lauren Gillespie
cs197.stanford.edu | cs197c.stanford.edu

Slides adapted from previous iterations of the course by Michael Bernstein and Jingyi Li

Administrivia

197

You all have started to work on your projects!

Due Thurs 4/27, 10am:

Assignment 3 — Introduction.

Progress Report I:

Get started on your projects: some lightweight setup work determined by your CA

197C

You have your research milestone plan and can start working!

Due Thurs 4/27, 10am:

Assignment 2, Part 4: Write up RW

Research milestone: **mentor sign-off**!

Administrivia

197: Project members dropped?

Don't worry! We will re-scope project + grading expectations so you're not affected.

197: Newcomers

We will give extensions to make up Week 1 and 2 assignments — please check with your CA!

Please check syllabus for grade breakdown, logistics, attendance policy, etc.

Anonymous Feedback

Option 1: High-resolution course evaluation (polled by email)

Option 2: Go to course website; scroll to bottom

SUBMIT COURSE FEEDBACK

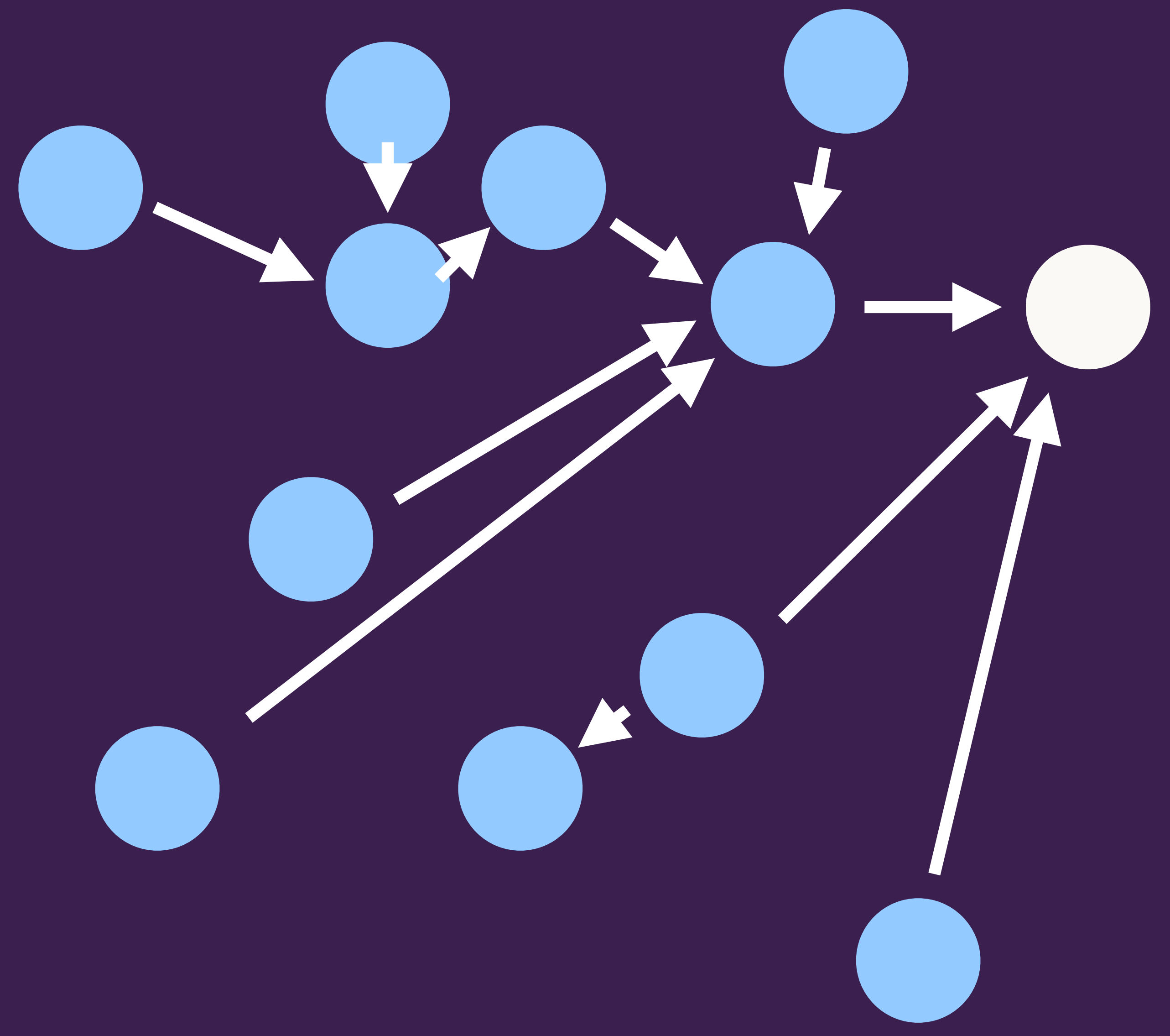
Anonymous Feedback Form

For problems with this site, contact the course staff email.

Last time

Literature Search

Check for novelty

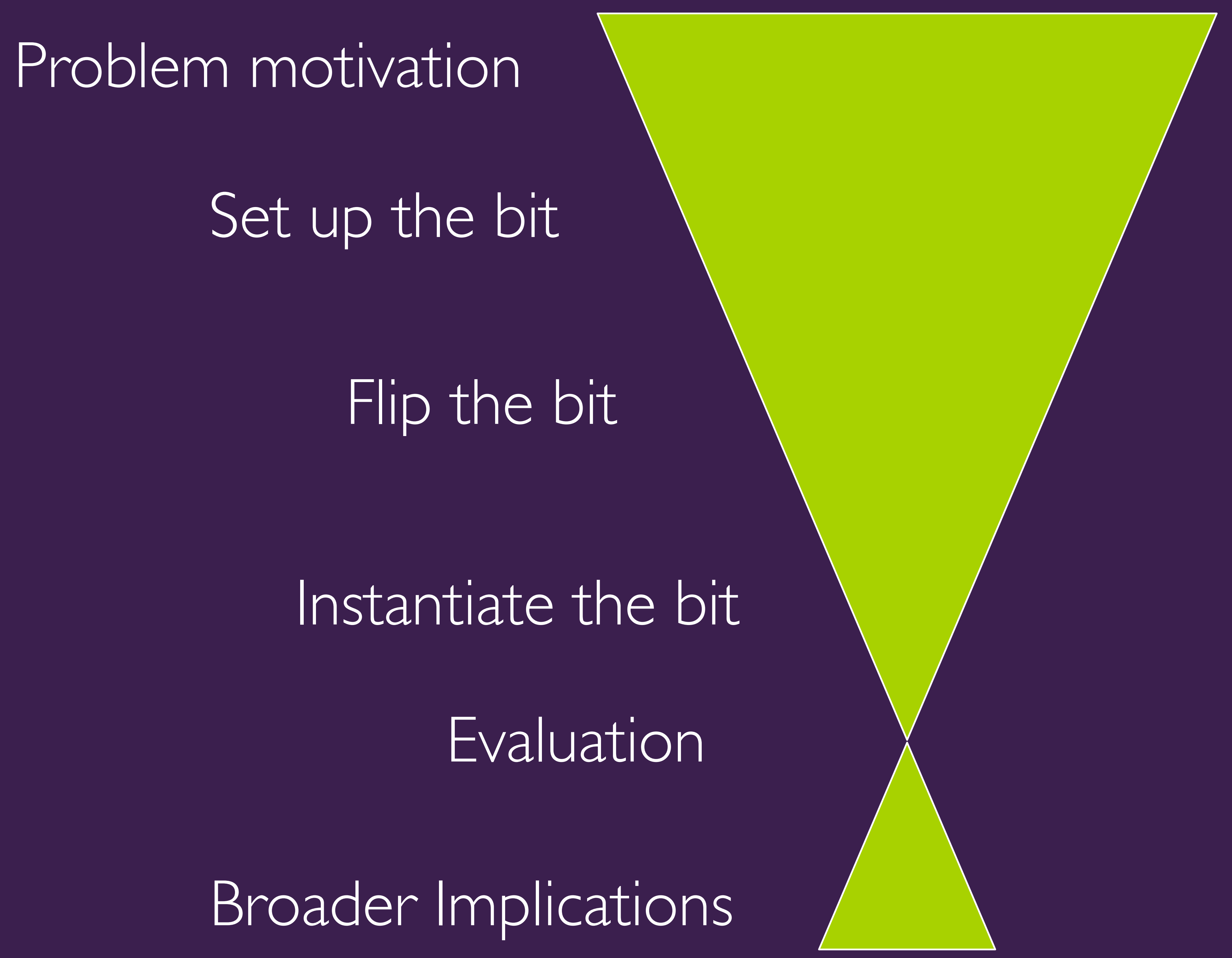


Related Work: Argue for novelty

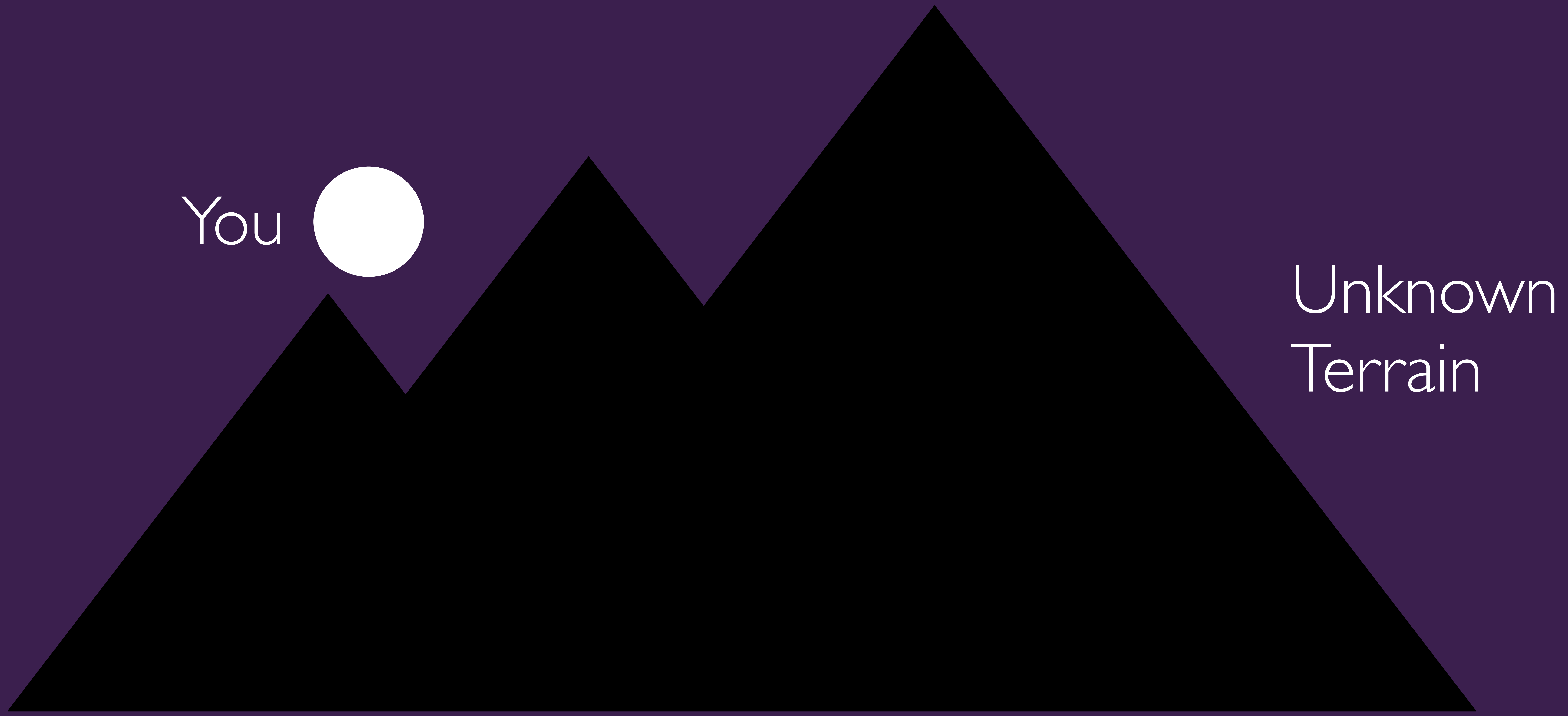
$$X \longrightarrow X'$$

Arguing a Research Project

Introduction: Articulate your problem + solution



Today: How do we do research?



Today: How do we do research?

Example

Problem

Activity tracking has important health applications, but it requires custom hardware.

Bit Flip

We can use just a standard cell phone for activity tracking.

More accessible and can be just as effective.

Instantiate the bit

We propose X, a prototype that can track walking and jogging.

But how?

Example

Problem

Audio classification and retrieval have important applications. CLIP shows promising results on learning image, text embeddings.

Bit flip

Add an audio embedding space to CLIP and leverage the power of CLIP for audio-related downstream tasks

Instantiate the bit

We add an X audio encoder and train with a new trio loss between (image, text, audio).

↓

But how?

What problem are we solving?

"But how do we start?"

"I'm feeling so lost."

"I thought of an important reason that this won't work."

"It's not working yet. I'm not sure that we're making progress."

Today’s big idea: vectoring

What is vectoring?

How do we vector effectively?

What goes wrong if we don’t vector?

Today: Vectoring

Vectoring: identifying the biggest dimension of risk in your project right now

today

Velocity: rapid reduction of risk in the chosen dimension

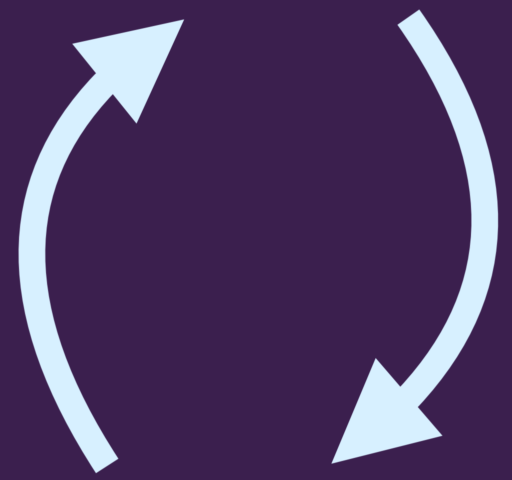
not today!

Bernstein theory of faculty success

To be a Stanford-tier faculty member, you need to master two skills that operate in a tight loop with one another.

Vectoring: identifying the biggest dimension of risk in your project right now

today



Velocity: rapid reduction of risk in the chosen dimension

not today!

What Is Vectoring?

And why do we need it?

What research is not

1. Figure out what to do.
2. Do it.
3. Publish.

What research is

Research is an iterative process of exploration, not a linear path from idea to result [Gowers 2000]

Problematic points of view

"OK, we have a good idea. Let's build it / model it / prove it / get training data."

"I spent some time thinking about this and hacking on it, and it's not going to work: it has a fatal flaw."

→ **Treating your research goal as a project spec and executing it**

Idea as project spec

Taking a concept and trying to realize it in parallel across all decisions, assumptions, and goals

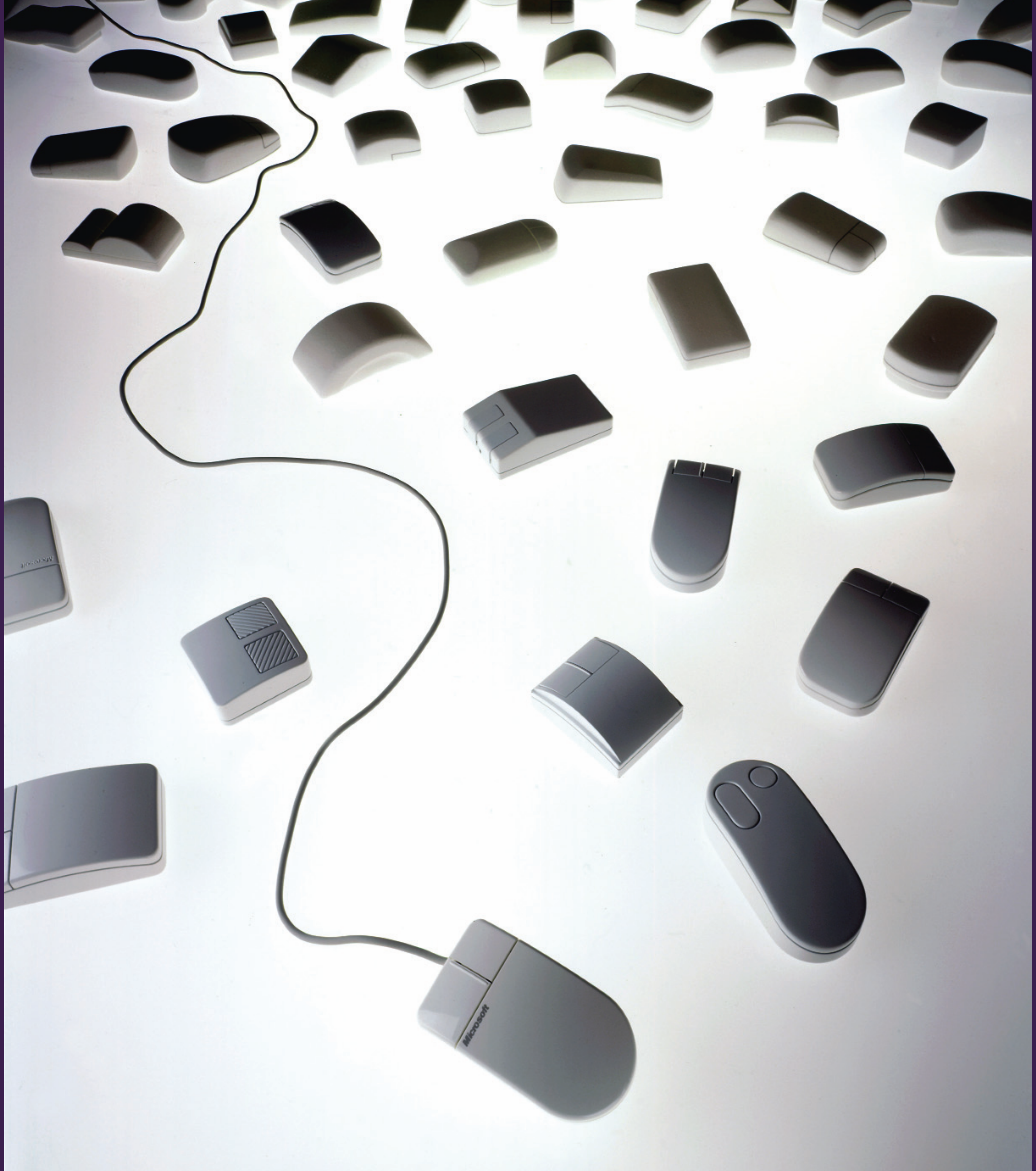
Why iterative?

Pick one vector

The more dimensions there are, the harder gradient descent becomes.

Instead of trying to do everything at once (project spec), pick one dimension of uncertainty — one vector — and focus on reducing its risk and uncertainty.

Scope your vector to be something you can reduce uncertainty on in 1–2 weeks


Microsoft

Example vectors

Piloting: will this technique work at all? To answer this, we implement a basic version of the technique and mock in the data and other test harness elements.

Engineering: will this technique work with a realistic workload? To answer this, we need to engineer a test harness.

Proving: does the limit exist that I suspect does? To answer this, we start by writing a proof for a simpler case.

Design: what might this interaction look like to an end user? To answer this, we create a low-fi prototype.

Implications

The vectors under consideration will each imply building different parts of your system.

Rather than building them all at once, when you might have to change things later, vectoring instead implies that you start by reducing uncertainty in the most important dimension first — your "inner loop" — and then building out from there.

Vectoring algorithm

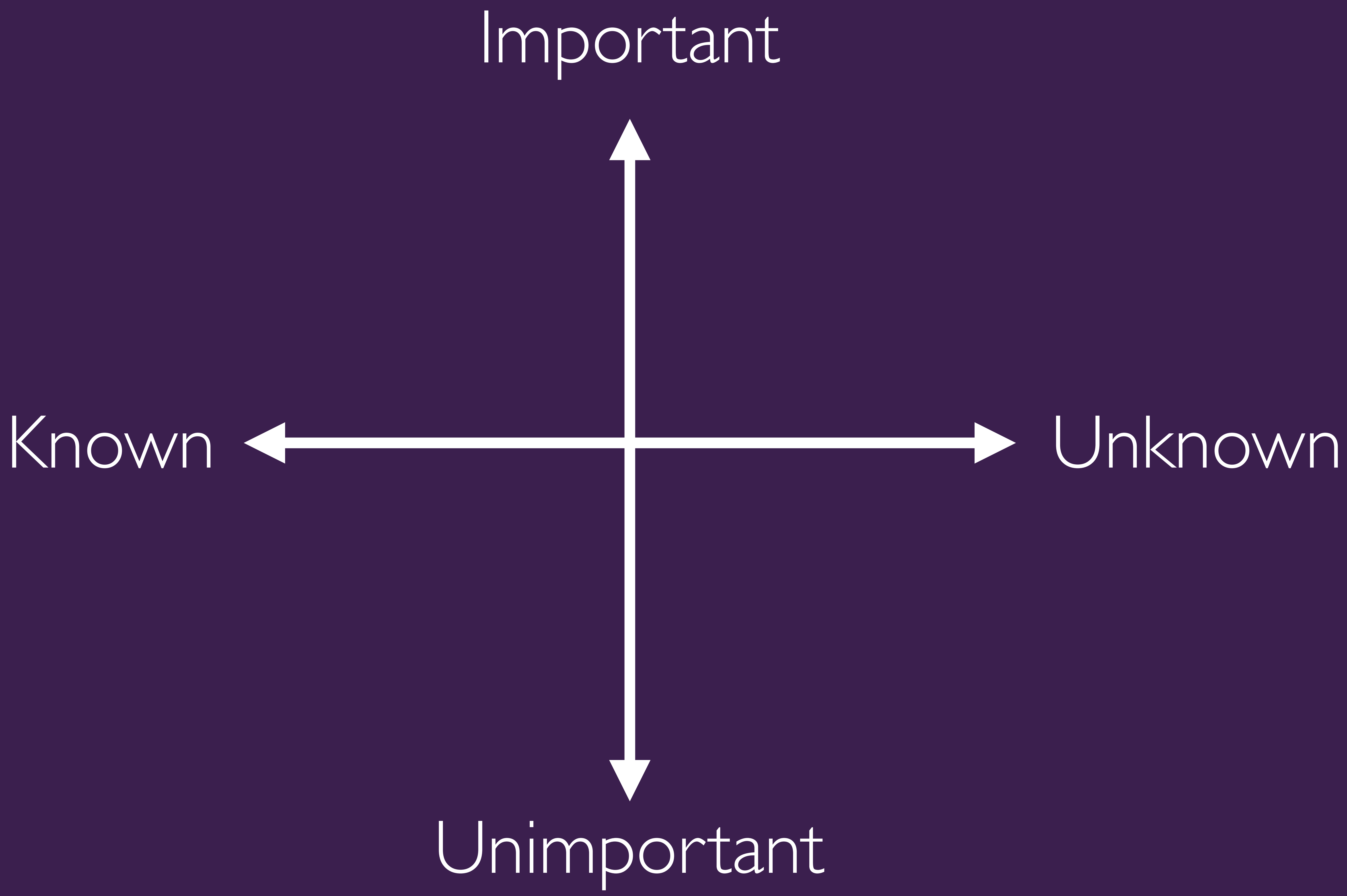
1. Generate questions
Untested hunches, risky decisions,
high-level directions

2. Rank your questions
Which is most critical?

3. Pick one and answer it rapidly
Answer only the most critical question
(This is where velocity comes into play)

Assumption mapping

Assumption mapping is a strategy for articulating questions and ranking them.



Let’s Try It

Learning

We thought that, in domains where ML still cannot succeed, we could draw on crowdsourcing to identify human-labeled predictive features. In other words, that people are great at identifying potentially informative features, but might be poor at weighing those features correctly to arrive at a prediction.

What's our first step?

Flock: Hybrid Crowd-Machine Learning Classifiers

Justin Cheng and Michael S. Bernstein
Stanford University
{jcccf, msb}@cs.stanford.edu

ABSTRACT
We present hybrid crowd-machine learning classifiers: classification models that start with a written description of a learning goal, use the crowd to suggest predictive features and label data, and then weigh these features using machine learning to produce models that are accurate and use human-understandable features. These hybrid classifiers enable fast prototyping of machine learning models that can improve on both algorithm performance and human judgment, and accomplish tasks where automated feature extraction is not yet feasible. *Flock*, an interactive machine learning platform, instantiates this approach. To generate informative features, Flock asks the crowd to compare paired examples, an approach inspired by analogical encoding. The crowd's efforts can be focused on specific subsets of the input space where machine-extracted features are not predictive, or instead used to partition the input space and improve algorithm performance in subregions of the space. An evaluation on six prediction tasks, ranging from detecting deception to differentiating impressionist artists, demonstrated that aggregating crowd features improves upon both asking the crowd for a direct prediction and off-the-shelf machine learning features by over 10%. Further, hybrid systems that use both crowd-nominated and machine-extracted features can outperform those that use either in isolation.

Author Keywords
Crowdsourcing, interactive machine learning

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Identifying predictive features is key to creating effective machine learning classifiers. Whether the task is link prediction or sentiment analysis, and no matter the underlying model, the "black art" of feature engineering plays a critical role in success [10]. Feature engineering is largely domain-specific, and users of machine learning systems spend untold hours experimenting. Often, the most predictive features only emerge

CSCW 2015, March 14–18, 2015, Vancouver, BC, Canada.

ACM 978-1-4503-2922-4/15/03 ...$15.00.
http://dx.doi.org/10.1145/2675133.2675214

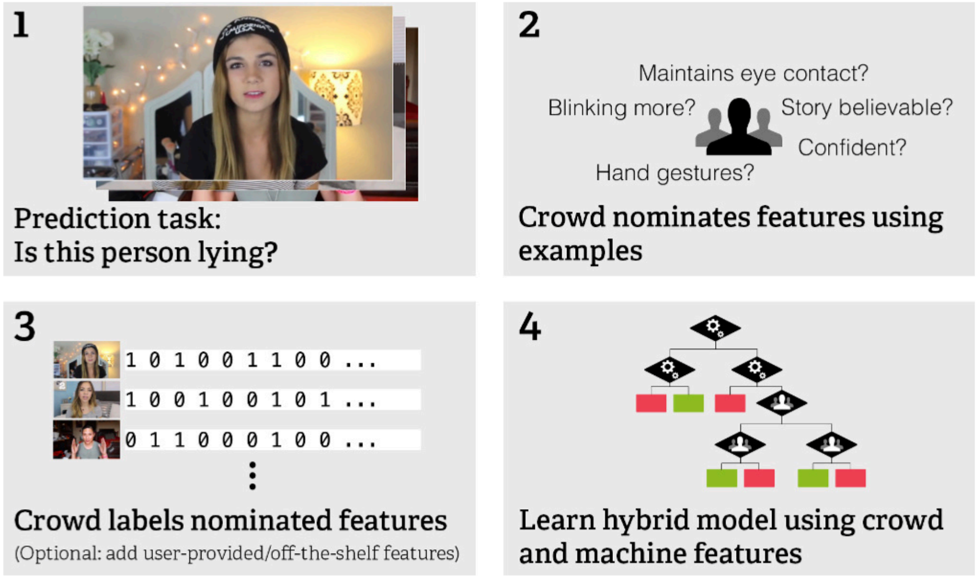


Figure 1. Flock is a hybrid crowd-machine learning platform that capitalizes on analogical encoding to guide crowds to nominate effective features, then uses machine learning techniques to aggregate their labels.

after many iterations [36]. And though feature engineers may have deep domain expertise, they are only able to incorporate features that are extractable via code.

However, *embedding crowds inside of machine learning architectures* opens the door to hybrid learners that can explore feature spaces that are largely unreachable by automatic extraction, then train models that use human-understandable features (Figure 1). Doing so enables fast prototyping of classifiers that can exceed both machine and expert performance. In this paper, we demonstrate classifiers that identify people who are lying, perform quality assessment of Wikipedia articles, and differentiate impressionist artists who use similar styles. Previous work that bridges crowdsourcing and machine learning has focused on optimizing the crowd's efforts (e.g., [8, 21, 39]): we suggest that inverting the relationship and embedding crowd insight inside live classifiers enables machine learning to be deployed for new kinds of tasks.

We present *Flock*, an end-user machine learning platform that uses paid crowdsourcing to speed up the prototyping loop and augment the performance of machine learning systems. Flock contributes a model for creating hybrid classifiers that intelligently embed both crowd and machine features. The system allows users to rapidly author hybrid crowd-machine learners by structuring a feature nomination process using the crowd, aggregating the suggested features, then collecting labels on these new features. It loops and gathers more crowd features to improve performance on subsets of the space where the model is misclassifying many examples. For instance, given a decision tree that uses machine-readable features, Flock can dynamically grow subtrees from nodes that have high classification error, or even replace whole branches. In addition to

Learning

Possible vectors:

- Can people identify predictive features for a single domain, e.g., lie detection?
- Can people estimate which features are going to be informative?
- Would a hybrid classifier (human features and labels as input to an ML model) actually perform well?

Flock: Hybrid Crowd-Machine Learning Classifiers

Justin Cheng and Michael S. Bernstein
Stanford University
{jcccf, msb}@cs.stanford.edu

ABSTRACT

We present hybrid crowd-machine learning classifiers: classification models that start with a written description of a learning goal, use the crowd to suggest predictive features and label data, and then weigh these features using machine learning to produce models that are accurate and use human-understandable features. These hybrid classifiers enable fast prototyping of machine learning models that can improve on both algorithm performance and human judgment, and accomplish tasks where automated feature extraction is not yet feasible. *Flock*, an interactive machine learning platform, instantiates this approach. To generate informative features, Flock asks the crowd to compare paired examples, an approach inspired by analogical encoding. The crowd's efforts can be focused on specific subsets of the input space where machine-extracted features are not predictive, or instead used to partition the input space and improve algorithm performance in subregions of the space. An evaluation on six prediction tasks, ranging from detecting deception to differentiating impressionist artists, demonstrated that aggregating crowd features improves upon both asking the crowd for a direct prediction and off-the-shelf machine learning features by over 10%. Further, hybrid systems that use both crowd-nominated and machine-extracted features can outperform those that use either in isolation.

Author Keywords

Crowdsourcing, interactive machine learning

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Identifying predictive features is key to creating effective machine learning classifiers. Whether the task is link prediction or sentiment analysis, and no matter the underlying model, the "black art" of feature engineering plays a critical role in success [10]. Feature engineering is largely domain-specific, and users of machine learning systems spend untold hours experimenting. Often, the most predictive features only emerge after many iterations [36]. And though feature engineers may have deep domain expertise, they are only able to incorporate features that are extractable via code.

CSCW 2015, March 14–18, 2015, Vancouver, BC, Canada.

ACM 978-1-4503-2922-4/15/03 ...$15.00.
http://dx.doi.org/10.1145/2675133.2675214

Figure 1. Flock is a hybrid crowd-machine learning platform that capitalizes on analogical encoding to guide crowds to nominate effective features, then uses machine learning techniques to aggregate their labels.

However, *embedding crowds inside of machine learning architectures* opens the door to hybrid learners that can explore feature spaces that are largely unreachable by automatic extraction, then train models that use human-understandable features (Figure 1). Doing so enables fast prototyping of classifiers that can exceed both machine and expert performance. In this paper, we demonstrate classifiers that identify people who are lying, perform quality assessment of Wikipedia articles, and differentiate impressionist artists who use similar styles. Previous work that bridges crowdsourcing and machine learning has focused on optimizing the crowd's efforts (e.g., [8, 21, 39]): we suggest that inverting the relationship and embedding crowd insight inside live classifiers enables machine learning to be deployed for new kinds of tasks.

We present *Flock*, an end-user machine learning platform that uses paid crowdsourcing to speed up the prototyping loop and augment the performance of machine learning systems. Flock contributes a model for creating hybrid classifiers that intelligently embed both crowd and machine features. The system allows users to rapidly author hybrid crowd-machine learners by structuring a feature nomination process using the crowd, aggregating the suggested features, then collecting labels on these new features. It loops and gathers more crowd features to improve performance on subsets of the space where the model is misclassifying many examples. For instance, given a decision tree that uses machine-readable features, Flock can dynamically grow subtrees from nodes that have high classification error, or even replace whole branches. In addition to

Teaming

We wanted to create an algorithm that would weave collaboration networks to help spread ideas over time by moving people from team to team.

What's our first step?

Hive: Collective Design Through Network Rotation

NILOUFAR SALEHI, UC Berkeley, USA
MICHAEL S. BERNSTEIN, Stanford University, USA

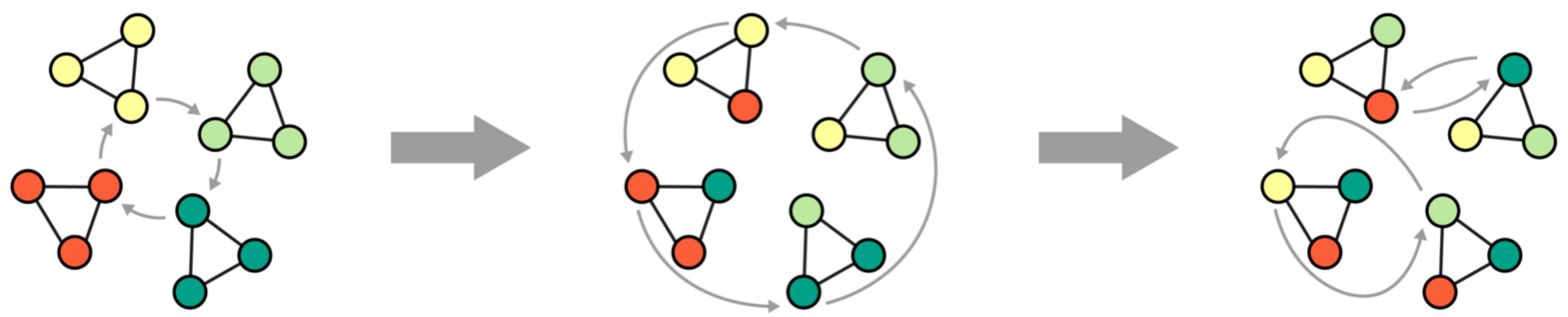


Fig. 1. Hive facilitates engagement with diverse viewpoints by rotating team membership in a collective over time. We introduce algorithmically-mediated *network rotation* to manage who should move, and when, to bring positive external influence to a team.

Collectives gather online around challenges they face, but frequently fail to envision shared outcomes to act on together. Prior work has developed systems for improving collective ideation and design by exposing people to each others' ideas and encouraging them to intermix those ideas. However, organizational behavior research has demonstrated that intermixing ideas does not result in meaningful engagement with those ideas. In this paper, we introduce a new class of collective design system that intermixes *people* instead of *ideas*: instead of receiving mere exposure to others' ideas, participants engage deeply with other members of the collective who represent those ideas, increasing engagement and influence. We thus present Hive: a system that organizes a collective into small teams, then intermixes people by rotating team membership over time. At a technical level, Hive must balance two competing forces: (1) networks are better at connecting diverse perspectives when network efficiency is high, but (2) moving people diminishes tie strength within teams. Hive balances these two needs through *network rotation*: an optimization algorithm that computes who should move where, and when. A controlled study compared network rotation to alternative rotation systems which maximize only tie strength or network efficiency, finding that network rotation produced higher-rated proposals. Hive has been deployed by Mozilla for a real-world open design drive to improve Firefox accessibility.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Design; online collaboration; participatory design; teams.

ACM Reference Format:
Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 151 (November 2018), 26 pages. https://doi.org/10.1145/3274420

Authors' addresses: Niloufar Salehi, UC Berkeley, School of Information, South Hall, Berkeley, CA, 94720, USA, nsalehi@berkeley.edu; Michael S. Bernstein, Stanford University, 353 Serra Mall, Stanford, CA, 94305, USA, msb@cs.stanford.edu.

Teaming

Possible vectors:

Do new members with new perspectives actually exert influence in practice?

If we prioritize or de-prioritize membership rotation in a simple (greedy) algorithm, does it lead to different outcomes in the collaboration network?

Hive: Collective Design Through Network Rotation

NILOUFAR SALEHI, UC Berkeley, USA
MICHAEL S. BERNSTEIN, Stanford University, USA

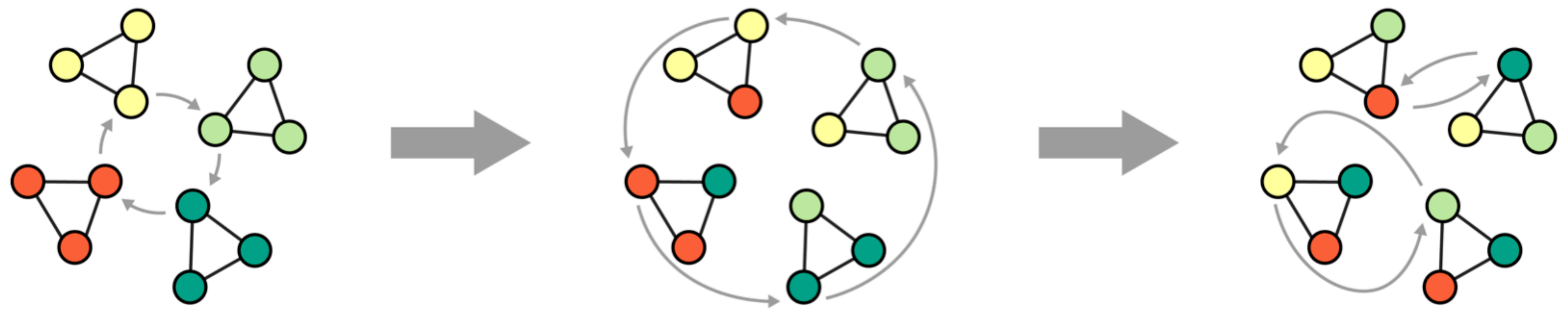


Fig. 1. Hive facilitates engagement with diverse viewpoints by rotating team membership in a collective over time. We introduce algorithmically-mediated *network rotation* to manage who should move, and when, to bring positive external influence to a team.

Collectives gather online around challenges they face, but frequently fail to envision shared outcomes to act on together. Prior work has developed systems for improving collective ideation and design by exposing people to each others' ideas and encouraging them to intermix those ideas. However, organizational behavior research has demonstrated that intermixing ideas does not result in meaningful engagement with those ideas. In this paper, we introduce a new class of collective design system that intermixes *people* instead of *ideas*: instead of receiving mere exposure to others' ideas, participants engage deeply with other members of the collective who represent those ideas, increasing engagement and influence. We thus present Hive: a system that organizes a collective into small teams, then intermixes people by rotating team membership over time. At a technical level, Hive must balance two competing forces: (1) networks are better at connecting diverse perspectives when network efficiency is high, but (2) moving people diminishes tie strength within teams. Hive balances these two needs through *network rotation*: an optimization algorithm that computes who should move where, and when. A controlled study compared network rotation to alternative rotation systems which maximize only tie strength or network efficiency, finding that network rotation produced higher-rated proposals. Hive has been deployed by Mozilla for a real-world open design drive to improve Firefox accessibility.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Design; online collaboration; participatory design; teams.

ACM Reference Format:
Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 151 (November 2018), 26 pages. https://doi.org/10.1145/3274420

Authors' addresses: Niloufar Salehi, UC Berkeley, School of Information, South Hall, Berkeley, CA, 94720, USA, nsalehi@berkeley.edu; Michael S. Bernstein, Stanford University, 353 Serra Mall, Stanford, CA, 94305, USA, msb@cs.stanford.edu.

Photo Editing

We wanted to create an algorithm that would allow users to resize objects in photographs.

What's our first step?

ZoomShop: Depth-Aware Editing of Photographic Composition

Sean J. Liu[1], Maneesh Agrawala[1], Stephen DiVerdi[2] and Aaron Hertzmann[2]

[1]Stanford University
[2]Adobe Research

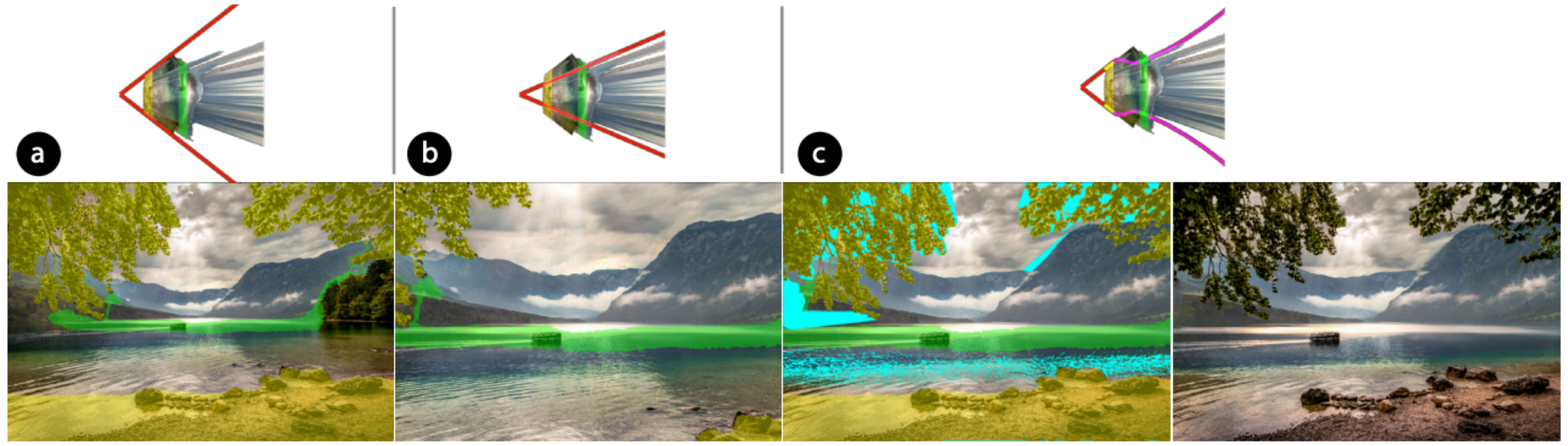


Figure 1: *Using ZoomShop to edit a photograph [Tha16]. (a) An image of mountains (background) framed by trees (foreground). The user's goal is to make the boat bigger while keeping the framing of foreground trees. (b) Zooming in and cropping scales up the boat, but cuts out most of the trees and shore. (c) Left: With ZoomShop users can select depth ranges (yellow and green) and independently adjust each region while maintaining scene structure. Disoccluded and stretched pixels are shown in cyan. Right: Cyan pixels from image are manually inpainted using Photoshop's Content-Aware Fill. We show the top-down view volume and boundary curve of each camera above the corresponding images in a, b, and c.*

Abstract

We present ZoomShop, a photographic composition editing tool for adjusting relative size, position, and foreshortening of scene elements. Given an image and corresponding depth map as input, ZoomShop combines a novel non-linear camera model and a depth-aware image warp to reproject and deform the image. Users can isolate objects by selecting depth ranges and adjust their scale and foreshortening, which controls the paths of the camera rays through the scene. Users can also select 2D image regions and translate them, which determines the objective function in the image warp optimization. We demonstrate that ZoomShop can be used to achieve useful compositional goals, such as making a distant object more prominent while preserving foreground scenery, or making objects both larger and closer together so they still fit in the frame.

CCS Concepts

• ***Computing methodologies*** → ***Graphics systems and interfaces; Image manipulation;***

1. Introduction

An important task in photographic composition is to adjust relative object sizes and positions. Consider Figure 1a, a photograph of a boat on a lake framed by trees. The photographer may wish to make the boat appear larger, which can be achieved by zooming, but the trees and the shore go out of frame (Fig. 1b). A longer lens from further back might satisfy both goals, but may be impossible due to physical constraints. With varying degrees of manual effort, today's digital tools allow adjusting the sizes of scene objects in a plausible, if not geometrically accurate, way [Mat21]. There are many methods that can arbitrarily adjust object sizes [AS07; CAA09; SLL19; BSFG09; SDM19], but these methods can fail to preserve important spatial relationships because they do not incorporate 3D scene understanding.

We present ZoomShop, a digital image editing tool that uses knowledge of the image's 3D geometry to provide the specific

Photo Editing

Possible vectors:

- Do cut-and-paste methods work?
- Do naive 2D warping methods work?
- What are the perceptual features we need to maintain, given the distortions that this task introduces?

ZoomShop: Depth-Aware Editing of Photographic Composition

Sean J. Liu[1], Maneesh Agrawala[1], Stephen DiVerdi[2] and Aaron Hertzmann[2]

[1]Stanford University
[2]Adobe Research

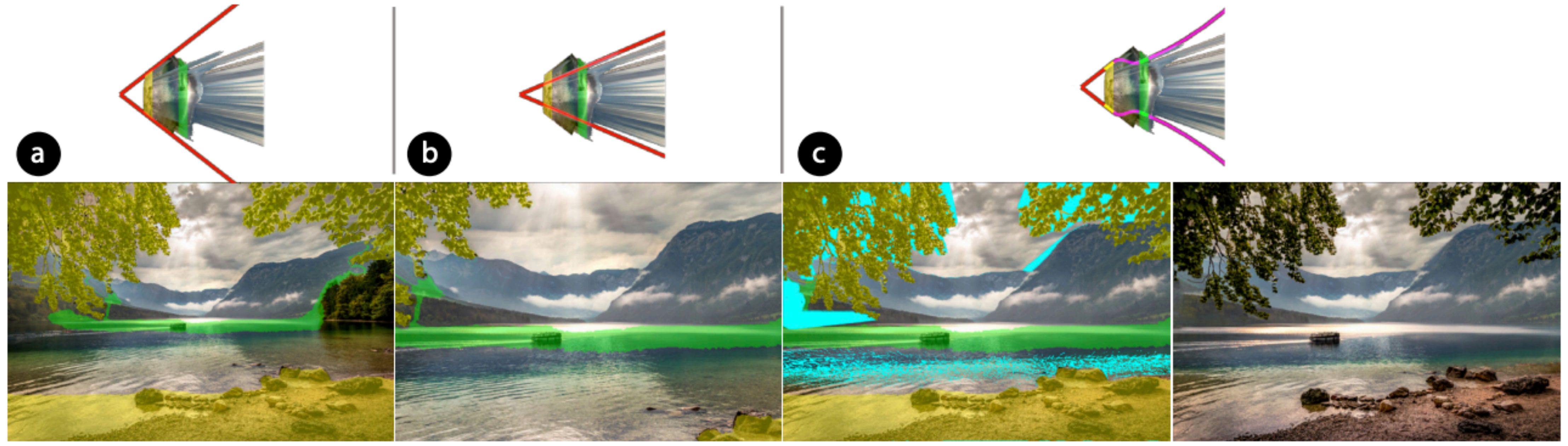


Figure 1: *Using ZoomShop to edit a photograph [Tha16]. (a) An image of mountains (background) framed by trees (foreground). The user's goal is to make the boat bigger while keeping the framing of foreground trees. (b) Zooming in and cropping scales up the boat, but cuts out most of the trees and shore. (c) Left: With ZoomShop users can select depth ranges (yellow and green) and independently adjust each region while maintaining scene structure. Disoccluded and stretched pixels are shown in cyan. Right: Cyan pixels from image are manually inpainted using Photoshop's Content-Aware Fill. We show the top-down view volume and boundary curve of each camera above the corresponding images in a, b, and c.*

Abstract

We present ZoomShop, a photographic composition editing tool for adjusting relative size, position, and foreshortening of scene elements. Given an image and corresponding depth map as input, ZoomShop combines a novel non-linear camera model and a depth-aware image warp to reproject and deform the image. Users can isolate objects by selecting depth ranges and adjust their scale and foreshortening, which controls the paths of the camera rays through the scene. Users can also select 2D image regions and translate them, which determines the objective function in the image warp optimization. We demonstrate that ZoomShop can be used to achieve useful compositional goals, such as making a distant object more prominent while preserving foreground scenery, or making objects both larger and closer together so they still fit in the frame.

CCS Concepts
• ***Computing methodologies*** → ***Graphics systems and interfaces; Image manipulation;***

1. Introduction

An important task in photographic composition is to adjust relative object sizes and positions. Consider Figure 1a, a photograph of a boat on a lake framed by trees. The photographer may wish to make the boat appear larger, which can be achieved by zooming, but the trees and the shore go out of frame (Fig. 1b). A longer lens from further back might satisfy both goals, but may be impossible due to physical constraints. With varying degrees of manual effort, today's digital tools allow adjusting the sizes of scene objects in a plausible, if not geometrically accurate, way [Mat21]. There are many methods that can arbitrarily adjust object sizes [AS07; CAA09; SLL19; BSFG09; SDM19], but these methods can fail to preserve important spatial relationships because they do not incorporate 3D scene understanding.

We present ZoomShop, a digital image editing tool that uses knowledge of the image's 3D geometry to provide the specific

VR Video

We wanted to create an authoring tool that seamlessly loops the VR video based on where the viewer is looking.

What's our first step?

View-Dependent Video Textures for 360° Video

Sean J. Liu **Maneesh Agrawala**
Stanford University
{lsean, maneesh}@cs.stanford.edu

Stephen DiVerdi **Aaron Hertzmann**
Adobe Research
{diverdi, hertzman}@adobe.com

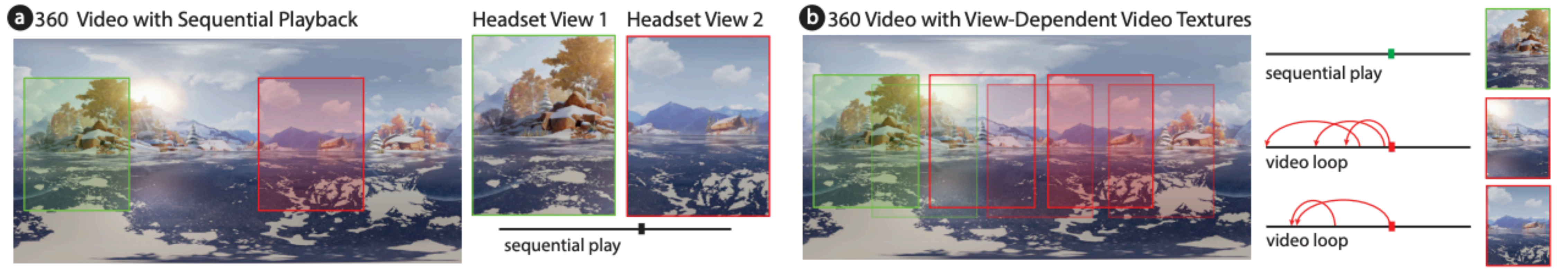


Figure 1. In 360° video, viewers can look anywhere at any time. In the opening scene of Invasion!, a rabbit emerges from a cave (a). In sequential playback, a viewer looking at the cave (green box) will see the rabbit emerge, whereas a viewer not looking at the cave (red box) will miss this event. We provide tools to guarantee that viewers see the region of interest (ROI) at the correct timecode to witness the event (b). We introduce the concept of gated clips, where playback only continues if the viewer satisfies a condition related to the ROI (green boxes). Otherwise, our player loops the video using view-dependent video textures (red boxes).

ABSTRACT
A major concern for filmmakers creating 360° video is ensuring that the viewer does not miss important narrative elements because they are looking in the wrong direction. This paper introduces **gated clips** which do not play the video past a gate time until a filmmaker-defined viewer gaze condition is met, such as looking at a specific region of interest (ROI). Until the condition is met, we seamlessly loop video playback using **view-dependent video textures**, a new variant of standard video textures that adapt the looping behavior to the portion of the scene that is within the viewer's field of view. We use our desktop GUI to edit live action and computer animated 360° videos. In a user study with casual viewers, participants prefer our looping videos over the standard versions and are able to successfully see all of the looping videos' ROIs without fear of missing important narrative content.

CCS Concepts
•**Human-centered computing** → **Virtual reality;** •**Computing methodologies** → **Computer vision problems;**

Author Keywords
view-dependent video texture, 360° video, virtual reality, cinematography, gaze guidance

INTRODUCTION
The medium of 360° video provides new artistic opportunities for filmmakers, allowing them to create videos with a greater sense of immersion and engagement than with conventional video. It also presents new challenges. In traditional cinematography, the director has full control over the camera orientation, field of view, zoom, and focus at all times. Traditional filmmakers use these controls to drive the narrative, ensuring that the viewer sees each important story element at the right time. With 360° videos, however, directors no longer have this control, and viewers can look in any direction at any time. As a result, viewers may miss important story content and become lost or confused as the story progresses.

For example, in the animated short "Invasion!" [7], the story begins with an establishing shot placing the viewer in the middle of an icy lake (Figure 1). Initially, the viewer is given time to look around and become familiar with their surroundings. A rabbit eventually emerges from a small cave. However, if the viewer is not looking at the cave entrance when the rabbit emerges, they will not see the rabbit, or what it does next. A second type of example is in the "Stranger Things: The VR Experience" [33] short film. As the tension rises, the viewer answers the phone, and is told to turn around. Then, a monster attacks from the direction opposite the phone. If the viewer does not turn around fast enough, they miss the monster attacking. A third type of example occurs in "Wild: The Experience" [14], where the viewer is placed between a hiker and an empty rock, on which a "ghost" appears only if the viewer is not looking at the rock. The hiker and the sound of her breathing is intended to get the viewer's attention away from the rock, so that the ghost can appear outside the viewer's field of view. However, if the viewer never looks away from the rock, the video player will reach the end of the video without the ghost ever appearing. Although the directors of these

VR Video

Possible vectors:

- How does naive looping perform?
- Do viewers like watching videos that wait for them by looping?

View-Dependent Video Textures for 360° Video

Sean J. Liu **Maneesh Agrawala**
Stanford University
{lsean, maneesh}@cs.stanford.edu

Stephen DiVerdi **Aaron Hertzmann**
Adobe Research
{diverdi, hertzman}@adobe.com

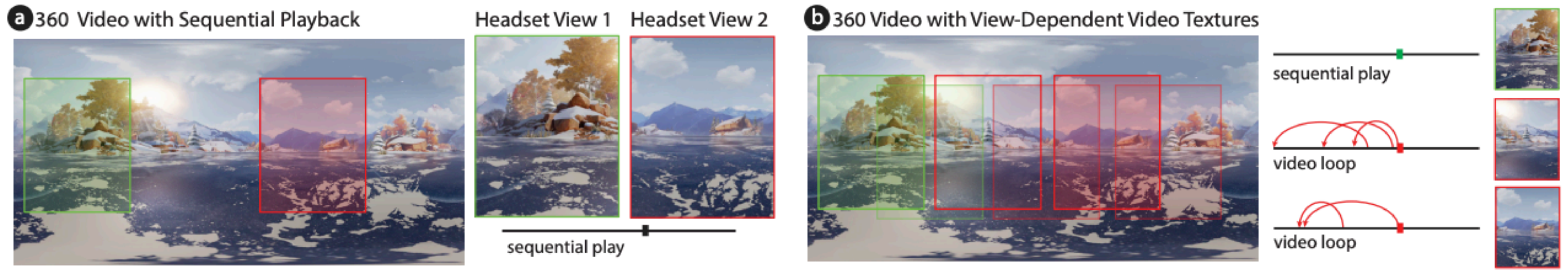


Figure 1. In 360° video, viewers can look anywhere at any time. In the opening scene of Invasion!, a rabbit emerges from a cave (a). In sequential playback, a viewer looking at the cave (green box) will see the rabbit emerge, whereas a viewer not looking at the cave (red box) will miss this event. We provide tools to guarantee that viewers see the region of interest (ROI) at the correct timecode to witness the event (b). We introduce the concept of gated clips, where playback only continues if the viewer satisfies a condition related to the ROI (green boxes). Otherwise, our player loops the video using view-dependent video textures (red boxes).

ABSTRACT

A major concern for filmmakers creating 360° video is ensuring that the viewer does not miss important narrative elements because they are looking in the wrong direction. This paper introduces **gated clips** which do not play the video past a gate time until a filmmaker-defined viewer gaze condition is met, such as looking at a specific region of interest (ROI). Until the condition is met, we seamlessly loop video playback using **view-dependent video textures**, a new variant of standard video textures that adapt the looping behavior to the portion of the scene that is within the viewer's field of view. We use our desktop GUI to edit live action and computer animated 360° videos. In a user study with casual viewers, participants prefer our looping videos over the standard versions and are able to successfully see all of the looping videos' ROIs without fear of missing important narrative content.

CCS Concepts

•Human-centered computing → Virtual reality; •Computing methodologies → Computer vision problems;

Author Keywords

view-dependent video texture, 360° video, virtual reality, cinematography, gaze guidance

INTRODUCTION

The medium of 360° video provides new artistic opportunities for filmmakers, allowing them to create videos with a greater sense of immersion and engagement than with conventional video. It also presents new challenges. In traditional cinematography, the director has full control over the camera orientation, field of view, zoom, and focus at all times. Traditional filmmakers use these controls to drive the narrative, ensuring that the viewer sees each important story element at the right time. With 360° videos, however, directors no longer have this control, and viewers can look in any direction at any time. As a result, viewers may miss important story content and become lost or confused as the story progresses.

For example, in the animated short "Invasion!" [7], the story begins with an establishing shot placing the viewer in the middle of an icy lake (Figure 1). Initially, the viewer is given time to look around and become familiar with their surroundings. A rabbit eventually emerges from a small cave. However, if the viewer is not looking at the cave entrance when the rabbit emerges, they will not see the rabbit, or what it does next. A second type of example is in the "Stranger Things: The VR Experience" [33] short film. As the tension rises, the viewer answers the phone, and is told to turn around. Then, a monster attacks from the direction opposite the phone. If the viewer does not turn around fast enough, they miss the monster attacking. A third type of example occurs in "Wild: The Experience" [14], where the viewer is placed between a hiker and an empty rock, on which a "ghost" appears only if the viewer is not looking at the rock. The hiker and the sound of her breathing is intended to get the viewer's attention away from the rock, so that the ghost can appear outside the viewer's field of view. However, if the viewer never looks away from the rock, the video player will reach the end of the video without the ghost ever appearing. Although the directors of these

Why is vectoring so important?

“If Ernest Hemingway, James Mitchener, Neil Simon, Frank Lloyd Wright, and Pablo Picasso could not get it right the first time, what makes you think that you will?”

— Paul Heckel

Iteration >> planning

Ideas rarely land exactly where you expect they will. It's best to test the most critical assumptions quickly, so that you can understand whether your hunch will play out, and what problems are worth spending time solving vs. kludging.

Human creative work is best in a loop of reflection and iteration. Vectoring is a way to make sure you're getting the most iteration cycles.

Re-vectoring

Often, after vectoring and reducing uncertainty in one dimension, it raises new questions and uncertainties.

In the next round of vectoring, you re-prioritize:

- If you get unexpected results and are confused (most of the time!), maybe it means you take a new angle to reduce uncertainty on a vector related to the prior one.
- If you answer your question to your own satisfaction (not completely, just to your satisfaction), you move on to the next most important vector

Magnitude of your vector

The result of vectoring should be something achievable in about a week's sprint. If it's not, you've picked too broad a question to answer.

> If your vectoring for "Can normal people be responsible for a lot of the trolling online?" is "Can normal people be responsible for a lot of the trolling on CNN.com?", you're still way too broad.
>
> That's evidence that you've just rescaled your project, ⟷ not picked a vector.

Core and Periphery

The output of a vectoring decision should allow you to identify what is **core** and what is **periphery** to reducing uncertainty in your vector of choice.

> **Core**: the goal that needs to be achieved in order to answer the question
>
> **Periphery**: the goals that can be faked, or assumed, or subsetted, or mocked in, so we can focus on the core.

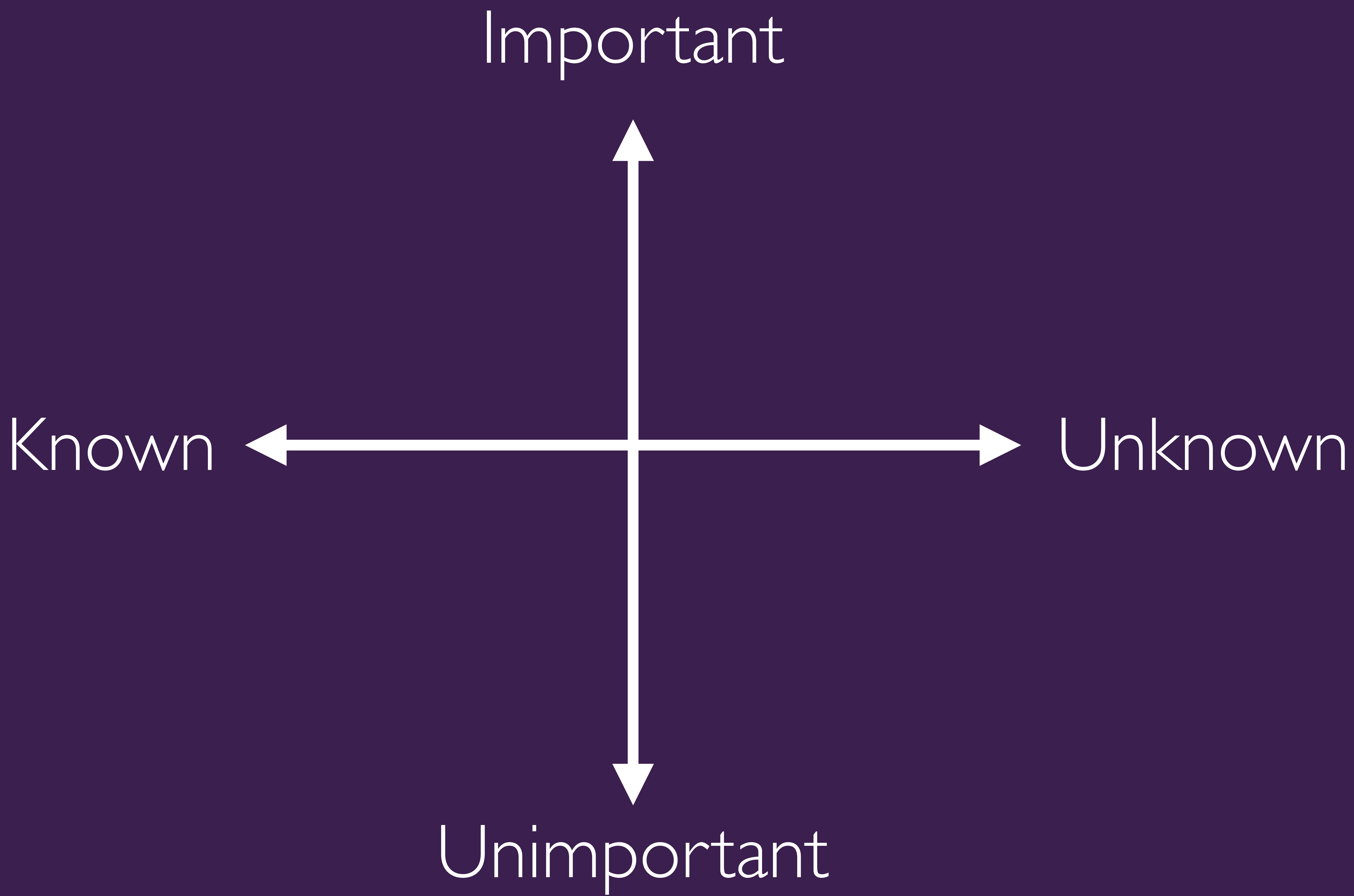
You should be able to make strong assumptions and use temporary scaffolding for anything that's periphery. (That's the velocity skill.)

Assumption mapping

Assumption mapping is a strategy for articulating questions and ranking them.

Try assumption mapping your project [5min]

(197: next week's vector)



Takeaways, in brief

1) The temptation is to try and solve the problem that’s set in front of you. Don’t.

2) Vectoring is a process of identifying the dimension of highest impact+uncertainty, and prioritizing that dimension while scaffolding the others

3) Successful vectoring enables you to rapidly hone in on the core insight of your research project

197

Due **next** Thurs 5/4: Assignment 4

Now, your project transitions to a state where your team is working to try and achieve the goal you set out in Assignment 3.

Each week for the next several weeks, your team will perform vectoring, submit a brief summary and slide, and report in section:

- This week's vector
- This week's plan
- This week's result
- Next week's vector

197C

Due **next** Thurs 5/4:

- Assignment 3: Introduction
- Start working towards your research milestone!

Reminder:

Submit your attendance on Canvas!

Vectoring in Research