

Writing a Paper & Picking Projects

CS 197 | Stanford University | Michael Bernstein

Administrivia

Reminder: Assignment 4 (Evaluation Plan) is due next Thursday

As part of this assignment, you will be working with staff to propose your plans through the end of the quarter.

Evaluation plan due week 8, Project reports through week 9, draft paper due in week 9, draft talk due in week 10, final paper and talk due during finals

Today's goals

We have a bunch of things we tried, some of them worked, some of them didn't — how do we write a paper about this?

Introducing the concept of **model papers** and how to use them

How do I pick projects to work on, going forward?

Writing A Paper

Scene Graph Prediction with Limited Labels

S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Ré, Li Fei-Fei
Stanford University
{vincentsc, paroma, ranjaykrishna, msb, chrmr, feifeili}@cs.stanford.edu

Abstract

knowledge bases such as Visual Genome power applications in computer vision, including visual answering and captioning, but suffer from sparse, e relationships. All scene graph models to date train on a small set of visual relationships thousands of training labels each. Hiring human

is expensive, and using textual knowledge base methods are incompatible with visual data. In

we introduce a semi-supervised method that assigns probabilistic relationship labels to a large number of images using few labeled examples. We analyze relationships to suggest two types of image-agnostic features are used to generate noisy heuristics, whose aggregated using a factor graph-based generative model creates enough training data to existing state-of-the-art scene graph model. We show that our method outperforms all baseline approaches on scene graph prediction by 5.16 recall@100 DCLS. In our limited label setting, we define a metric for relationships that serves as an indicator (0.778) for conditions under which our method over transfer learning, the de-facto approach for with limited labels.

Introduction

effort to formalize a structured representation for Visual Genome [27] defined **scene graphs**, a formalism similar to those widely used to represent knowledge [13, 18, 56]. Scene graphs encode objects (e.g., bike) as nodes connected via pairwise relationships (e.g., riding) as edges. This formalization has led to state-of-the-art models in image captioning [3], image understanding [25, 42], visual question answering [24], relation reasoning [26] and image generation [23]. However, scene graph models ignore more than 98% of object categories that do not have sufficient labeled instances (see Figure 2) and instead focus on modeling the



Figure 1. Our semi-supervised method automatically generates probabilistic relationship labels to train any scene graph model.

few relationships that have thousands of labels [31, 49, 54].

Hiring more human workers is an ineffective solution to labeling relationships because image annotation is so tedious that seemingly obvious labels are left unannotated. To complement human annotators, traditional text-based knowledge completion tasks have leveraged numerous semi-supervised or distant supervision approaches [16, 7, 17, 34]. These methods find syntactical or lexical patterns from a small labeled set to extract missing relationships from a large unlabeled set. In text, pattern-based methods are successful, as relationships in text are usually **document-agnostic** (e.g. <Tokyo - is capital of - Japan>). Visual relationships are often incidental: they depend on the contents of the particular image they appear in. Therefore, methods that rely on external knowledge or on patterns over concepts (e.g. most instances of dog next to frisbee are playing with it) do not generalize well. The inability to utilize the progress in text-based methods necessitates specialized methods for visual knowledge.

In this paper, we automatically generate missing relationships labels using a small, labeled dataset and use these generated labels to train downstream scene graph models (see Figure 1). We begin by exploring how to define **image-agnostic** features for relationships so they follow patterns across images. For example, eat usually consists of one object consuming another object smaller than itself, whereas look often consists of common objects: phone, laptop, or window (see Figure 3). These rules are not dependent on raw pixel values; they can be derived from image-agnostic features like object categories and relative spatial positions between objects in a relationship. While such rules are simple, their capacity to provide supervision for unannotated relationships has been unexplored. While image-agnostic

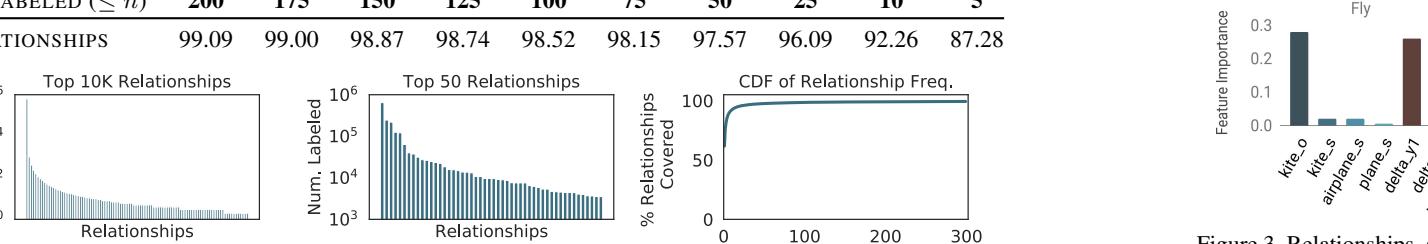


Figure 2. Visual relationships have a long tail (left) of infrequent relationships. Current models [49, 54] only focus on the top 50 relationships (middle) in the Visual Genome dataset, which all have thousands of labeled instances. This ignores more than 98% of the relationships with few labeled instances (right, top/table).

2. Related work

Textual knowledge bases were originally hand-curated by experts to structure facts [4, 5, 44] (e.g. <Tokyo - is capital of - Japan>). To scale dataset curation efforts, recent approaches mine knowledge from the web [9] or hire non-expert annotators to manually curate knowledge [5, 47].

In semi-supervised solutions, a small amount of labeled text is used to extract and exploit patterns in unlabeled sentences [2, 21, 33–35, 37]. Unfortunately, such approaches cannot be directly applied to visual relationships: textual relations can often be captured by external knowledge or patterns, while visual relationships are often local to an image.

Visual relationships have been studied as spatial priors [14, 16], co-occurrences [51], language statistics [28, 31, 53], and within entity contexts [29]. Scene graph prediction models have dealt with the difficulty of learning from incomplete knowledge, as recent methods utilize statistical motifs [54] or object-relationship dependencies [30, 49, 50, 55].

All these methods limit their inference to the top 50 most frequently occurring predicate categories and ignore those without enough labeled examples (Figure 2).

The de-facto solution for limited label problems is **transfer learning** [15, 52], which requires that the source domain used for pre-training follows a similar distribution as the target domain. In our setting, the source domain is a dataset of frequently-labeled relationships with thousands of examples [30, 49, 50, 55], and the target domain is a set of limited label relationships. Despite similar objects in source and target domains, we find that transfer learning has difficulty generalizing to new relationships. Our method does not rely on availability of a larger, labeled set of relationships; instead, we use a small labeled set to annotate the unlabeled set of images.

Our contributions are three-fold. (1) We introduce the first method to complete visual knowledge bases by finding missing visual relationships (Section 5.1). (2) We show the utility of our generated labels in training existing scene graph prediction models (Section 5.2). (3) We introduce a metric to characterize the complexity of visual relationships and show it is a strong indicator ($R^2 = 0.778$) for our semi-supervised method's improvements over transfer learning (Section 5.3).

To address the issue of gathering enough training labels for machine learning models, **data programming** has emerged as a popular paradigm. This approach learns to model imperfect labeling sources in order to assign training labels to unlabeled data. Imperfect labeling sources can come from crowdsourcing [10], user-defined heuristics [8, 43], multi-instance learning [22, 40], and distant supervision [19] or unlabeled D_U . We turn over the image-agnostic features, and a factor-graph based generative model [31] to learn probabilistic labels to the unlabeled data.

As discussed in Section 3, we want to leverage image-agnostic features to learn rules that annotate unlabeled relationships.

Our approach assigns probabilistic labels to a set D_U of unlabeled images in three steps: (1) we extract image-agnostic features from the objects in the labeled D_p and

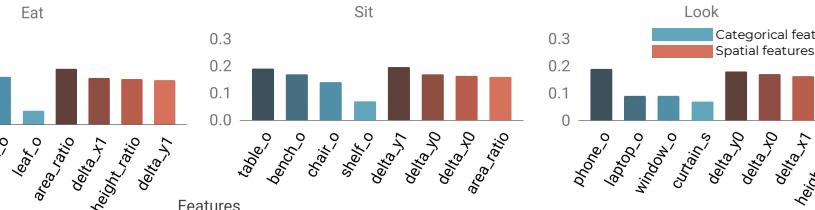


Figure 3. Relationships, such as fly, eat, and sit can be characterized effectively by their categorical (s and o refer to subject and object, respectively) or spatial features. Some relationships like fly rely heavily only on a few features — kites are often seen high up in the sky.

to label relationships with limited data? Previous literature has combined deep learning features with extra information extracted from categorical object labels and relative spatial object locations [25, 31]. We define categorical features, $\langle o, -, o' \rangle$, as a concatenation of one-hot vectors of the subject o and object o' . We define spatial features as:

$$\frac{x - x'}{w}, \frac{y - y'}{h}, \frac{(y + h) - (y' + h')}{w}, \\ \frac{(x + w) - (x' + w')}{w}, \frac{h'}{h}, \frac{w'}{w}, \frac{w' + h'}{wh}, \frac{w + h'}{w + h}$$

where $b = [y, x, h, w]$ and $b' = [y', x', h', w']$ are the top-left bounding box coordinates and their widths and heights.

To explore how well spatial and categorical features can describe different visual relationships, we train a simple decision tree model for each relationship. We plot the importances for the top 4 spatial and categorical features in Figure 3. Relationships like fly place high importance on the difference in y-coordinate between the subject and object, supporting probabilistic labels. Our approach achieves 47.53 recall@100 for predicate classification on Visual Genome, improving over the same model trained using only labeled instances by 40.97 points. For scene graph detection, our approach achieves within 8.65 recall@100 of the same model trained on the original Visual Genome dataset with 108x more labeled data. We end by comparing our approach to transfer learning, the de-facto choice for learning from limited labels.

3. Analyzing visual relationships

We define the formal terminology used in the rest of the paper and introduce the image-agnostic features that our semi-supervised method relies on. Then, we seek quantitative insights into how visual relationships can be described by the properties between its objects. We ask (1) what image-agnostic features can characterize visual relationships? and (2) given limited labels, how well do our chosen features characterize the complexity of relationships? With these in mind, we motivate our model design to generate heuristics that do not overfit to the small amount of labeled data and assign accurate labels to the larger, unlabeled set.

3.1. Terminology

A scene graph is a multi-graph G that consists of objects o as nodes and relationships r as edges. Each object $o_i = \{b_i, c_i\}$ consists of a bounding box b_i and its category $c_i \in \mathcal{C}$ where \mathcal{C} is the set of all possible object categories (e.g. dog, frisbee). Relationships are denoted $\langle \text{subject} - \text{predicate} - \text{object} \rangle$ or $\langle o - p - o' \rangle$. $p \in \mathbb{P}$ is a predicate, such as ride and eat. We assume that we have a small labeled set $\{(o, p, o') \in D_p\}$ of annotated relationships for each predicate p . Usually, these datasets are on the order of 10 examples or fewer. For our semi-supervised approach, we also assume that there exists a large set of images D_U without any labeled relationships.

3.2. Defining image-agnostic features

To understand the efficacy of image-agnostic features, we'd like to measure how well they can characterize the complexity of particular visual relationships. As seen in Figure 4, a visual relationship can be defined by a number of image-agnostic features (e.g. a person can ride a bike, or a dog can ride a surfboard). To systematically define this notion of complexity, we identify **subtypes** for each visual relationship. Each subtype captures one way that a relationship manifests in the dataset. For example, in Figure 4, ride contains one categorical subtype with <person - ride - bike> and another with <dog - ride - surfboard>. Similarly, a person might carry an object in different relative spatial orientations (e.g. on her head, to her side). As shown in Figure 5, visual relationships might have significantly different degrees of spatial and categorical complexity, and therefore a different number of subtypes for each. To compute spatial subtypes, we perform mean shift clustering [11] over the spatial features extracted from all the

3.3. Complexity of relationships

To understand the efficacy of image-agnostic features, we'd like to measure how well they can characterize the complexity of particular visual relationships. As seen in Figure 4, a visual relationship can be defined by a number of image-agnostic features (e.g. a person can ride a bike, or a dog can ride a surfboard). To systematically define this notion of complexity, we identify **subtypes** for each visual relationship. Each subtype captures one way that a relationship manifests in the dataset. For example, in Figure 4, ride contains one categorical subtype with <person - ride - bike> and another with <dog - ride - surfboard>. Similarly, a person might carry an object in different relative spatial orientations (e.g. on her head, to her side). As shown in Figure 5, visual relationships might have significantly different degrees of spatial and categorical complexity, and therefore a different number of subtypes for each. To compute spatial subtypes, we perform mean shift clustering [11] over the spatial features extracted from all the

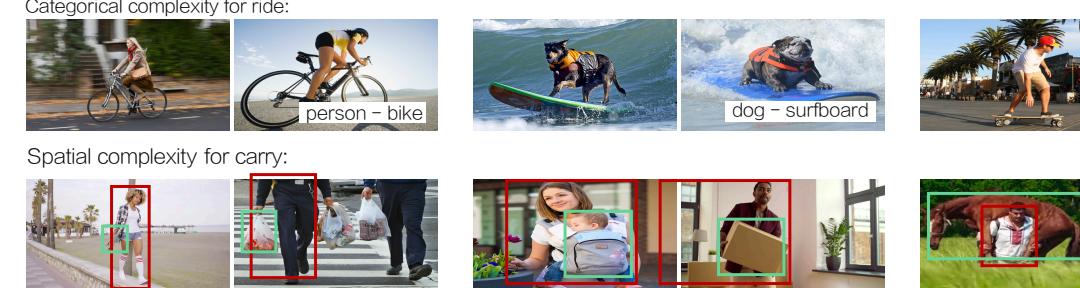


Figure 4. We define the number of subtypes of a relationship as a measure of its complexity. Subtypes can be categorized into Categorical complexity for ride: person - bike, dog - surfboard, and Spatial complexity for carry: person - ride - bike, dog - ride - surfboard.

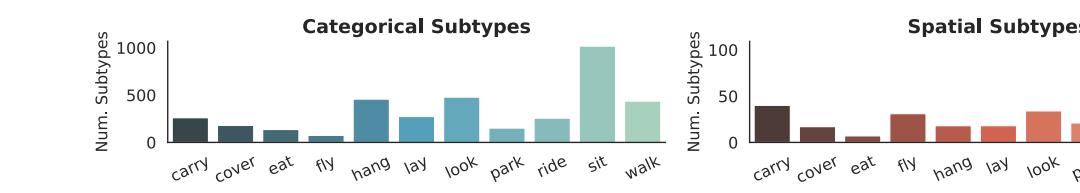


Figure 5. A subset of visual relationships with different levels of complexity as defined by spatial and categorical subtypes. we show how this measure is a good indicator of our semi-supervised method's effectiveness compared to baseline

Algorithm 1 Semi-supervised Alg. to train SGM

```

1: INPUT:  $\{(o, p, o') \in D_p\} \forall p \in \mathbb{P}$  — A small set with multi-class labels for predicates.
2: INPUT:  $\{(o, o') \in D_U\}$  — A large unlabeled set but no relationship labels.
3: INPUT:  $f(\cdot, \cdot)$  — A function that extracts features.
4: INPUT:  $DT(\cdot)$  — A decision tree.
5: INPUT:  $G(\cdot)$  — A generative model that assigns multiple labels for each datapoint.
6: INPUT: train( $\cdot$ ) — Function used to train a scene graph model.
7: Extract features and labels,  $X_p, Y_p := f(\{o, o'\} \in D_p)$ 
8:  $X_U := f(\{(o, o') \in D_U\} \in D_U)$ 
9: Generate heuristics by fitting J decision trees
10: Learn generic model  $G(A)$  and assign probabilities
11: Train scene graph model,  $SGM := \text{train}(D_p, G)$ 
12: OUTPUT:  $SGM(\cdot)$ 
```

from the object proposals extracted using Mask R-CNN [19] or unlabeled D_U . We turn over the image-agnostic features, and a factor-graph based generative model [31] to learn probabilistic labels to the unlabeled data.

Feature extraction: Our approach uses image-agnostic features defined in Section 3, which require a bounding box and category labels. The features are ground truth objects in D_p or from objects in D_U by running existing object detector [19].

Heuristic generation: We fit decision trees to the image-agnostic features from the objects in the labeled D_p and

we hypothesized earlier, TRANSFER LEARNING cases when the labeled set only capture the relationship's subtypes. This explains how Ours (CATEG. + SPAT.) given a small portion of labeled subtypes

4. Approach

We aim to automatically generate labels for missing visual relationships that can be then used to train any downstream scene graph prediction model. We assume that in the long-tail of infrequent relationships, we have a small labeled set $\{(o, p, o') \in D_p\}$ of annotated relationships for each predicate p (often, on the order of 10 examples or less). As discussed in Section 3, we want to leverage image-agnostic features to learn rules that annotate unlabeled relationships.

To address the issue of gathering enough training labels for machine learning models, **data programming** has emerged as a popular paradigm. This approach learns to model imperfect labeling sources in order to assign training labels to unlabeled data. Imperfect labeling sources can come from crowdsourcing [10], user-defined heuristics [8, 43], multi-instance learning [22, 40], and distant supervision [19] or unlabeled D_U .

We turn over the image-agnostic features, and a factor-graph based generative model [31] to learn probabilistic labels to the unlabeled data.

Feature extraction: Our approach uses image-agnostic features defined in Section 3, which require a bounding box and category labels. The features are ground truth objects in D_p or from objects in D_U by running existing object detector [19].

Heuristic generation: We fit decision trees to the image-agnostic features from the objects in the labeled D_p and

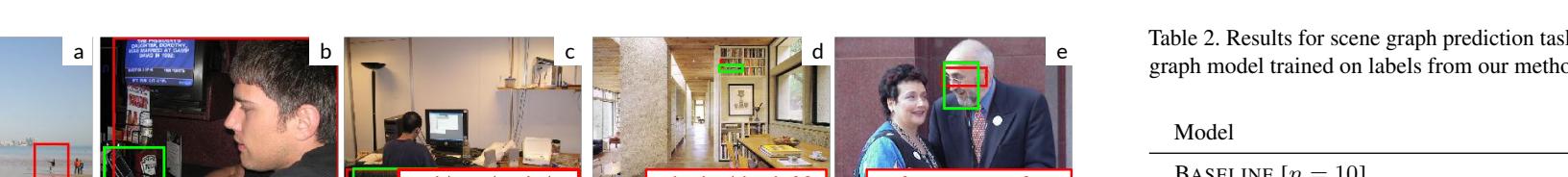
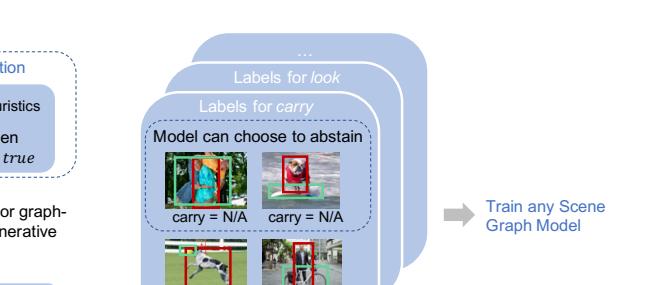


Figure 7. (a) Heuristics based on spatial features help predict <man - fly - kite>. (b) Our model learns that look is highly correlated with phone. (c) We overfit to the importance of chair as a categorical feature for sit, and fail to identify hang as the correct relationship. (d) We overfit to the spatial positioning associated with ride, where objects are typically longer and directly underneath the subject. (e) Given our image-agnostic features, we produce a reasonable label for <glasses - cover - face>. However, our model is incorrect, as two typically different predicates (sit and cover) share a semantic meaning in the context of <glasses - - - face>.

that our semi-supervised method outperforms transfer learning, which has seen more data. Furthermore, we quantify when our method outperforms transfer learning using our metric for measuring relationship complexity (Section 3.3). Eliminating synonyms and supersets. Typically, past scene graph approaches have used 50 predicates from Visual Genome to study visual relationships. Unfortunately, there are 50 treat synonyms like laying on and lying on as separate classes. To make matters worse, some predicates are considered supersets of others (i.e. above is a superset of above-right). This is problematic because the same predicate can have many different meanings between synonyms and supersets. For example, in this section, we eliminate all superpredicates and merge all synonyms, resulting in 20 unique predicates. In the Supplementary Material we include a list of these predicates and report our method's performance on all 50 predicates.

Dataset. We use two standard datasets, VRD [31] and Visual Genome [27], to evaluate on tasks related to visual relationships or scene graphs. Each scene graph contains objects localized as bounding boxes in the image along with pairwise relationships connecting them, categorized as action (e.g., carry), possessive (e.g., wear), spatial (e.g., above), or comparative (e.g., taller than) descriptors. In the Supplementary Material we include a list of these predicates and report our method's performance on all 50 predicates.

Method. These heuristics, individually, are noisy and do not assign labels to all object pairs in D_U . As a result, we aggregate the labels from all J heuristics. To do so, we use a factor graph-based generative model popularized by weak supervision techniques [1, 39, 41, 45, 48].

$$L_\theta = \mathbb{E}_{Y \sim \pi} [\log(1 + \exp(-\theta^T V^T Y))]$$

where θ is the learned parameters, π is the distribution learned by the generative model, Y is the true label, and V are features extracted by any scene graph prediction model.

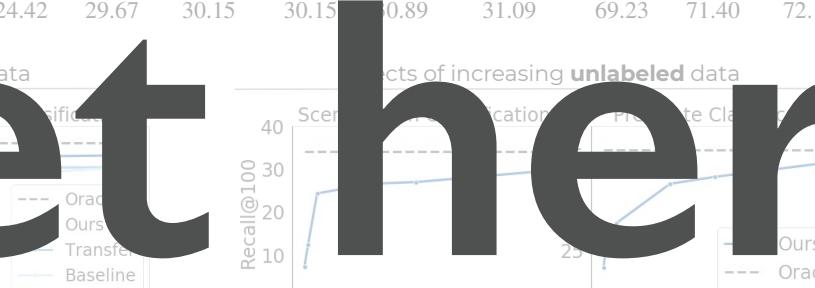


Figure 8. A scene graph model [54] trained using our labels outperforms both using TRANSFER LEARNING labels and using only the BASELINE labeled examples consistently across scene graph classification and predicate classification for different amounts of available labeled relationship instances. We also compare to ORACLE, which is trained with 108x more labeled data.

objects that have a large difference in y-coordinate

The common misunderstanding



work
work
work
coffee
work
work
imposter syndrome
work

Why is this a misunderstanding?

Research papers are complex documents, with too many degrees of freedom to “just write”. Being strategic will save time and avoid dead ends.

Scene Graph Prediction with Limited Labels

Vincent S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Ré, Li Fei-Fei
Stanford University
{vinentsc, paroma, ranjaykrishna, msb, chrismre, feifeili}@cs.stanford.edu

Abstract

Visual knowledge bases such as Visual Genome power numerous applications in computer vision, including visual question answering and captioning, but suffer from sparse, incomplete relationships. All scene graph models to date are limited to training on a small set of visual relationships that have thousands of training labels each. Hiring human annotators is expensive, and using textual knowledge base completion methods are incompatible with visual data. In this paper, we introduce a semi-supervised method that assigns probabilistic relationship labels to a large number of unlabeled images using few labeled examples. We analyze visual relationships to suggest two types of image-agnostic features that are used to generate noisy heuristics, whose outputs are aggregated using a factor graph-based generative model. With as few as 10 labeled examples per relationship, the generative model creates enough training data to train any existing state-of-the-art scene graph model. We demonstrate that our method outperforms all baseline approaches on scene graph prediction by 5.16 recall@100 for PREDCLS. In our limited label setting, we define a complexity metric for relationships that serves as an indicator ($R^2 = 0.778$) for conditions under which our method succeeds over transfer learning, the de-facto approach for training with limited labels.

1. Introduction

In an effort to formalize a structured representation for images, Visual Genome [27] defined **scene graphs**, a formalization similar to those widely used to represent knowledge bases [13,18,56]. Scene graphs encode objects (e.g. person, bike) as nodes connected via pairwise relationships (e.g., riding) as edges. This formalization has led to state-of-the-art models in image captioning [3], image retrieval [25,42], visual question answering [24], relationship modeling [26] and image generation [23]. However, all existing scene graph models ignore more than 98% of relationship categories that do not have sufficient labeled instances (see Figure 2) and instead focus on modeling the few relationships that have thousands of labels [31,49,54].

Hiring more human workers is an ineffective solution to labeling relationships because image annotation is so tedious that seemingly obvious labels are left unannotated. To complement human annotators, traditional text-based knowledge completion tasks have leveraged numerous semi-supervised or distant supervision approaches [5,7,17,31]. These methods find syntactical or lexical patterns from a small labeled set to extract missing relationships from a large unlabeled set. In text, pattern-based methods are successful, as relationships in text are usually **document-agnostic** (e.g. <Tokyo> -is capital of - Japan>). Visual relationships are often incidental: they depend on the contents of the particular image they appear in. Therefore, methods that rely on external knowledge or on patterns over concepts (e.g. most instances of dog next to frisbee are playing with it) do not generalize well. The inability to utilize the progress in text-based methods necessitates specialized methods for visual knowledge.

In this paper, we automatically generate missing relationships labels using a small, labeled dataset and use these generated labels to train downstream scene graph models (see Figure 1). We begin by exploring how to define **image-agnostic** features for relationships so they follow patterns across images. For example, eat usually consists of one object consuming another object smaller than itself, whereas look often consists of common objects: phone, laptop, or window (see Figure 3). These rules are not dependent on raw pixel values; they can be derived from image-agnostic features like object categories and relative spatial positions between objects in a relationship. While such rules are simple, their capacity to provide supervision for unannotated relationships has been unexplored. While image-agnostic

arXiv:1904.11622v2 [cs.CV] 20 Aug 2019

1

```
graph LR; L[Limited labels] --> O[Our semi-supervised method]; O --> P[Probabilistic training labels]; P --> S[Scene graph model]; S --> A[Any existing scene graph model]
```

Figure 1. Our semi-supervised method automatically generates probabilistic relationship labels to train any scene graph model.

...so what do we do instead?

There are many genres

Even within areas, there exist many different genres of paper. Each genre is typically built around the claim you are making, and implies a structure to the sections and to the writing. For example:

We solve a problem:
articulate the problem, explain what causes that problem and what others have done to deal with it, detail your approach, and prove that you make progress on the problem

We measure an outcome: explain that nobody has bothered understanding how a phenomenon behaves, explain how to create a study that sheds light, and report the outcomes of it

We introduce a technique: articulate a problem as above, but focus the narrative on the technique you've created, since it will generalize

Genres imply structure

Common “We Solve A Problem” structure:

Introduction: overview and thesis

Related Work: situate your contribution relative to prior research

Approach: describe your approach and important implementation details

Evaluation: test whether your approach succeeds at its stated goals

Method

Results

Discussion: reflect on limitations, implications, and future work

Conclusion: summarize and restate your contribution

*But, this will vary
by area!*

“Which genre is our project?”

You can often derive the appropriate genre in the same way that you derived the evaluation — what is the thesis and claim that you are supporting?

But this may be challenging until you've read a large number of papers. So instead...

Model papers

A model paper is a paper that you can use as a model or template for constructing your paper.

You should be able to structure your paper in the same way as your model paper

Follow its general flow of argument in the introduction

Use similar section and subsection heading organization

Create figures, tables, and graphs that fulfill the same function as theirs

Apply the same general proportions, e.g., number of pages per section

Selecting your model paper

Model paper != nearest neighbor paper

The model paper should be a paper that makes the same type of argument as yours. It should be in the same genre as you seek.

Often the nearest neighbor paper will make a similar form of argument, but not necessarily

Often the nearest neighbor paper will be a well-written paper, but not necessarily

Find your model paper and share it with your TA for a thumbs up before writing.

From model to paper

Start by reverse-outlining the model paper.

How does it structure its argument into sections?

What is the main expository goal of each section? What is its sub-thesis?

What role does each figure play?

From model to paper

Next, build a mapping from their outline to yours.

Translate each section and sub-section heading into what the equivalent heading is for you

Translate each sub-thesis into what the equivalent sub-thesis is for you

Translate each figure into what the equivalent figure is for you

What if it doesn't quite fit?

Model papers should be templates, not straightjackets. You will probably need to adapt your mapping slightly from what your model paper does.

e.g., you require a slightly different evaluation structure or visualization than them

e.g., you're drawing on a different literature than them, and need to explain something that they didn't

You can play with the genre — just don't discard the genre. Check with your TA for any substantial changes that you want to make.

Assignment 5: draft paper

Work together with your team to write a draft paper. This should be a complete draft in the template format of your research, and include reviewable drafts of every section.

“Can we include text we already wrote?” Absolutely! + tweaks

“Do we need the results of our evaluation?” Yes, but you can continue to update your results through the final deadline.

“What if our project doesn’t work out?” Still write up the report. Negative results can be valuable. Unpack in Discussion what it was about your idea or assumptions that wasn’t borne out.

After this, Assignment 6 will be a draft talk.

Picking Projects

Where do research ideas come from?

A common mindset: riffing

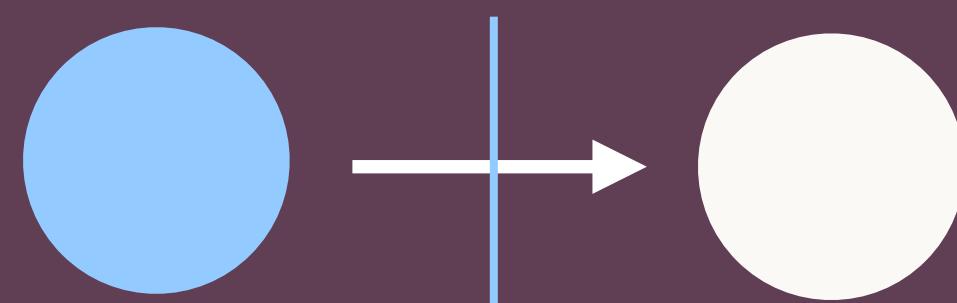
Ye Olde Riffing Recipe, from The Bernstein Cookbook for People Who Don't Cook Well But Can At Least Do Research:

Read a bunch of papers

Pick a paper you really like

Ask yourself: how could I extend this to another domain, or make progress on one of its challenging assumptions, or otherwise extend it?

This is a process for generating a one-paper bit flip



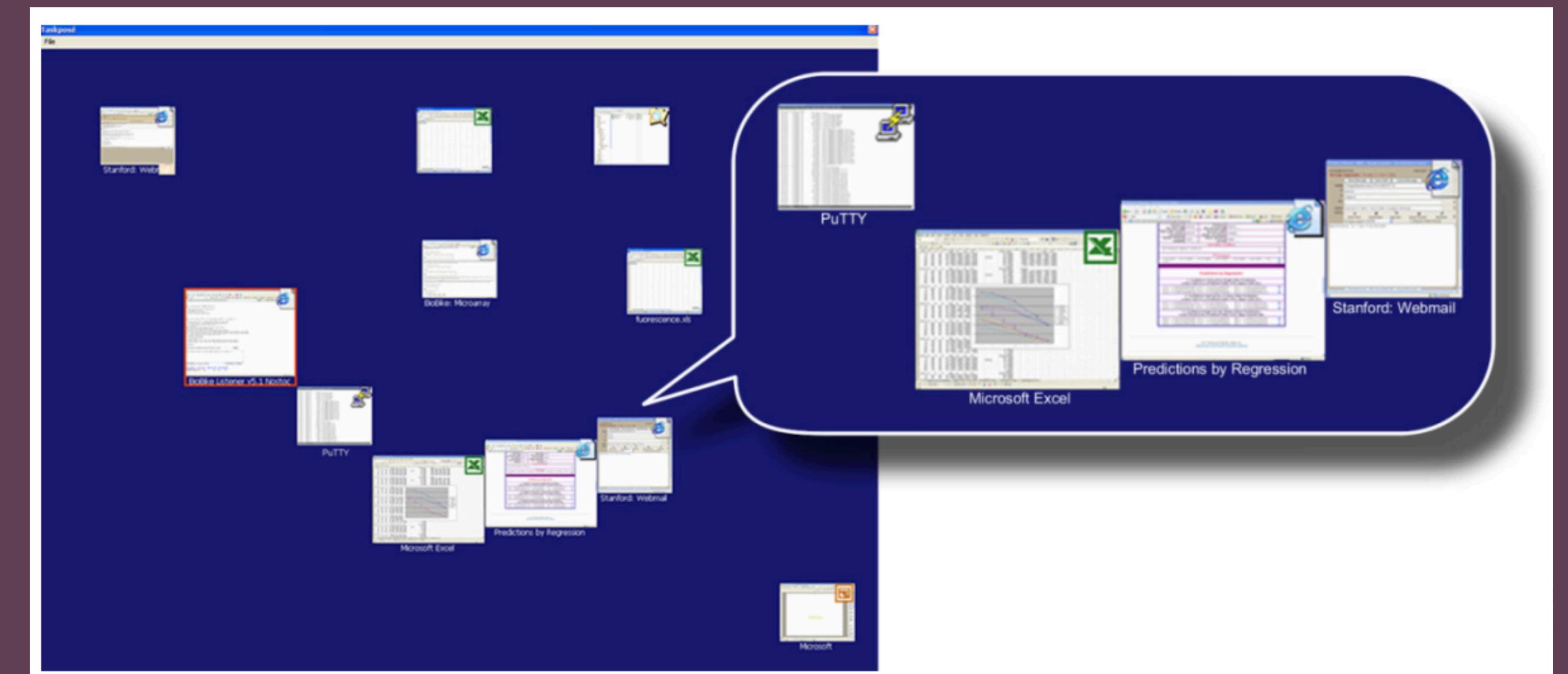
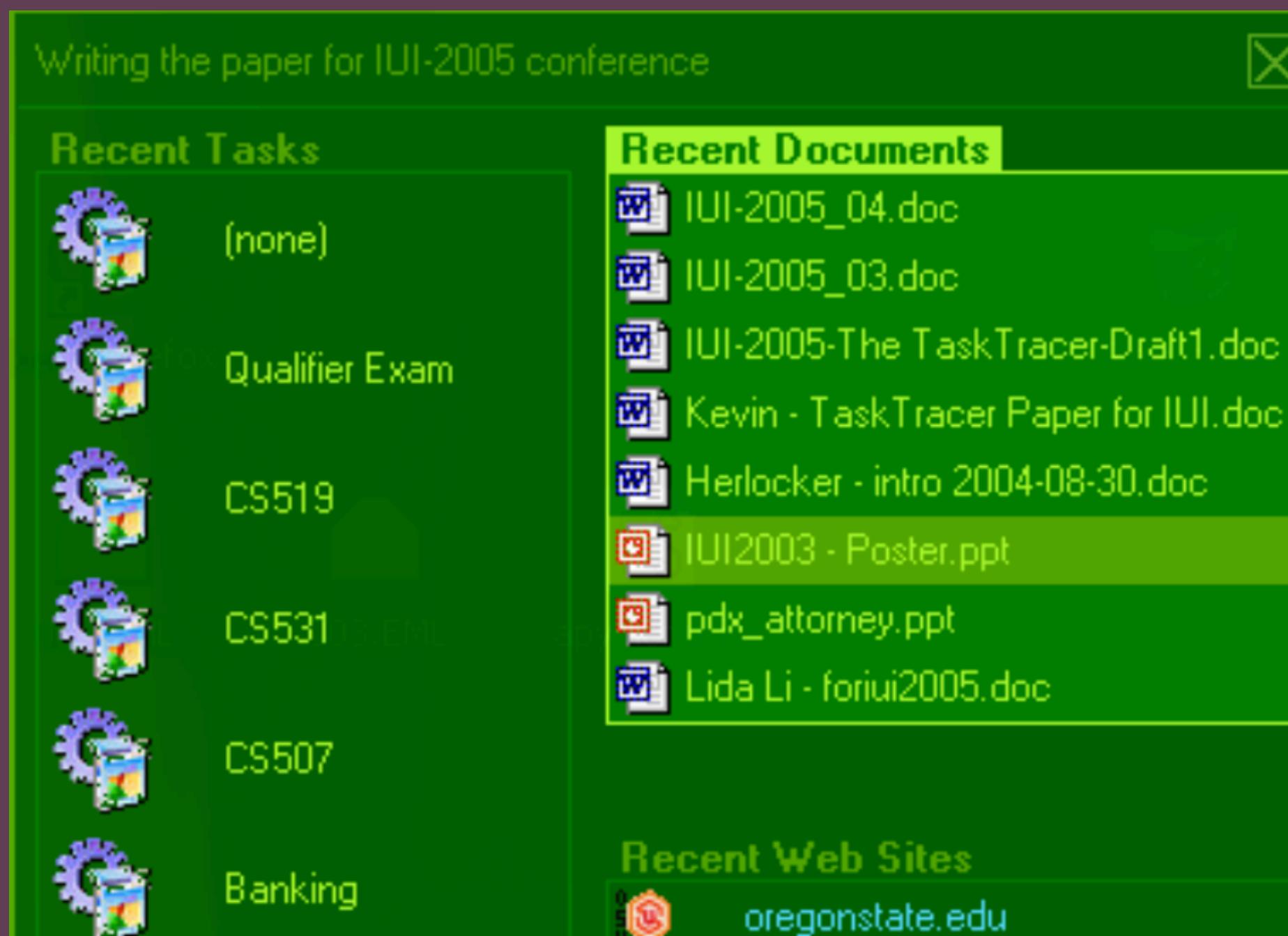
Riffing is often a good starting point for a first independent project

It places focus on execution, and gives you most of the inputs, outputs, and constraints—the assumptions—up front

Even me (here is my now-embarrassing first project)

Lots of work on task-centric
workspaces

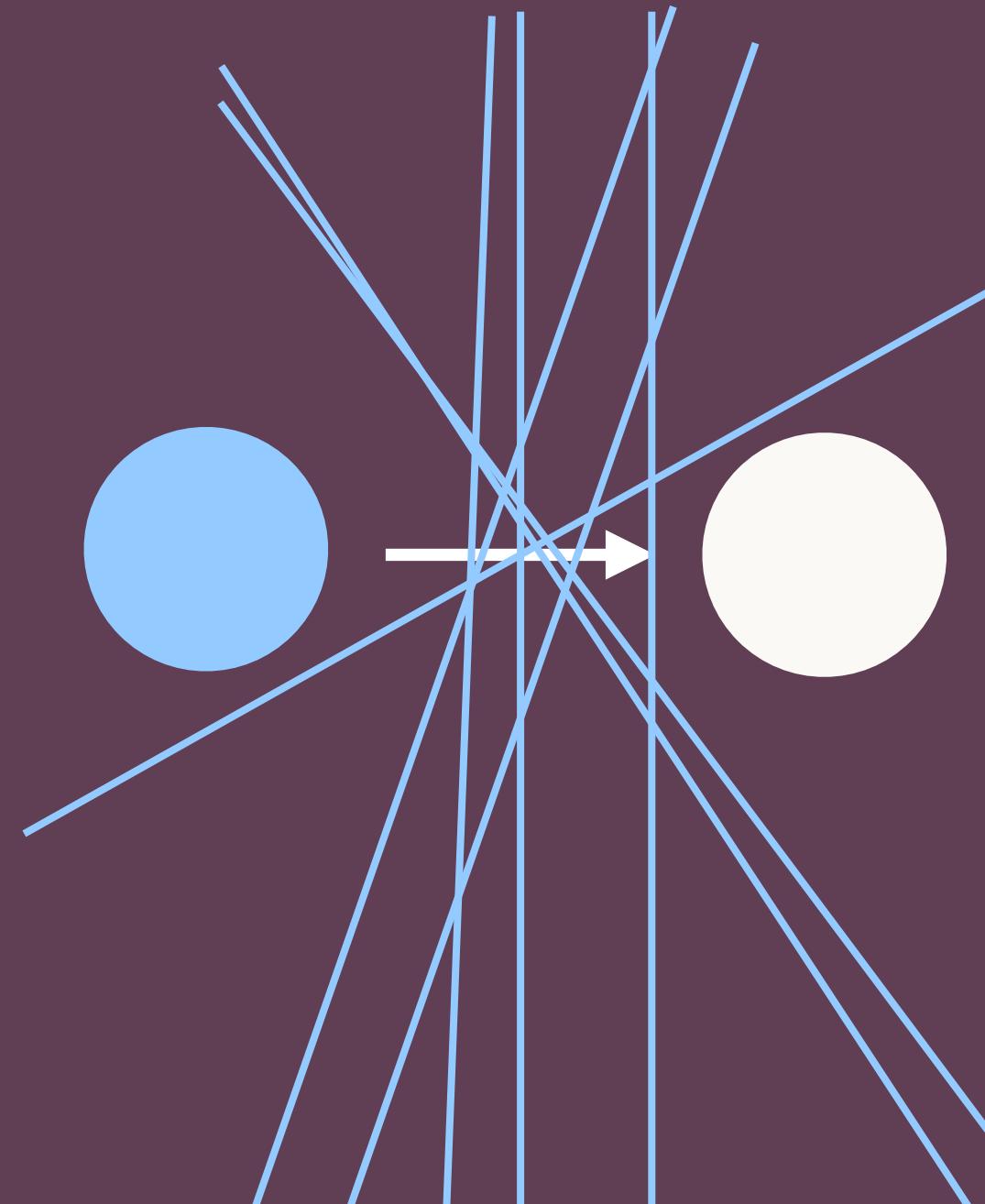
Me: “But tasks can have fuzzy boundaries!”



What are the risks here?

It's not clear that all bit flips are worthwhile.

My misappropriated quote: "Your scientists were so preoccupied with whether or not they could that they didn't stop to think if they should."



"Salami Science": possibility of incremental work when we don't view the field's assumptions broadly



What we mean when we say “incremental”

Research and science are not neutral: they embed values

Incrementally is a push back against minor adjustments to models that don't build substantial theory

What we mean when we say science isn't neutral

Science and Technology Studies (STS) establishes that what counts as a contribution, or as major vs. incremental, or even what counts as Computer Science, is socially constructed by elites in the field.

Not so long ago, HCI and Ethics were not seen as legitimate CS

Also not so long ago, CS itself was not seen as a legitimate field

Objection to creating a CS department at Stanford, via Leo Guibas:
“We don’t have a department of Refrigerator Science!”

Thanks to Jingyi Li!

So what should we do
instead of only riffing on
papers?

Desert Metaphor

Everyone look out, he's going to try and draw on the whiteboard.

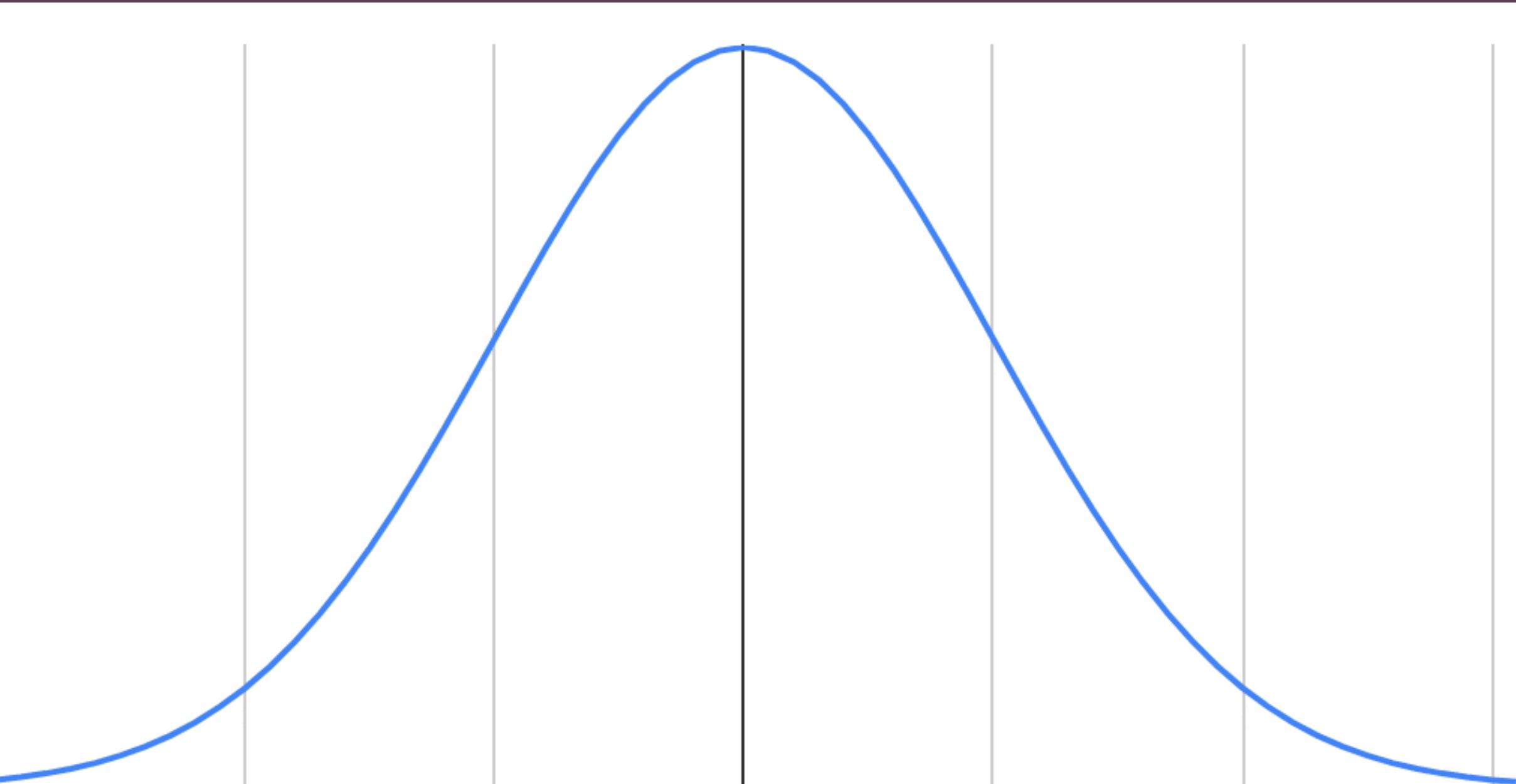
Is this a big rock?

Do I have an angle on it?

**“If you want to have a
good idea, you must
have many ideas.”**

– Nobel Prize winning chemist Linus Pauling

**“If you want to have a
good idea, you must
have many ideas.”**



$2 \cdot \sigma = 95\%$ of samples

$3 \cdot \sigma = 99.7\%$ of samples

Some Strategies and Stories

Rage-based research

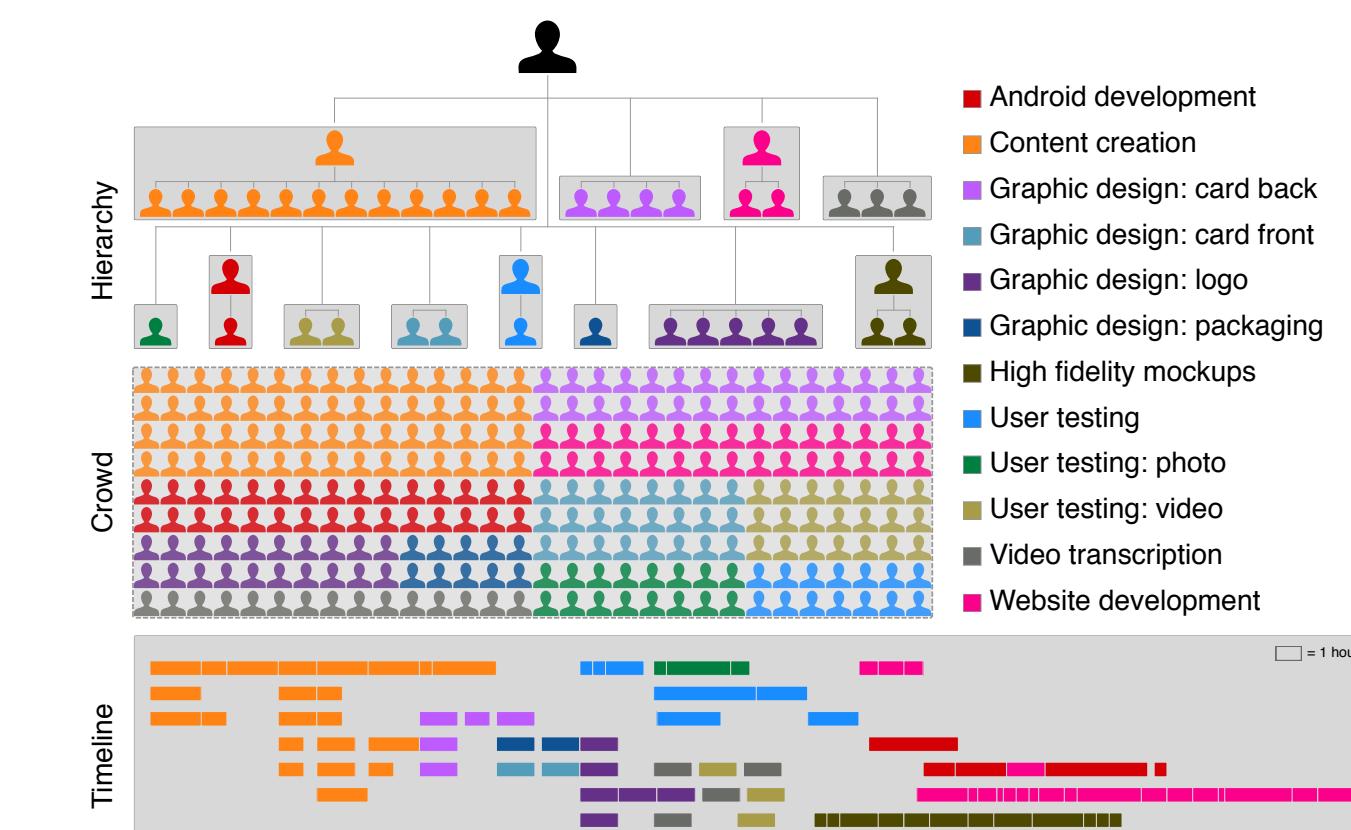
When a pattern or underlying assumption in the field starts to dig at you until you decide to prove that it's wrong.

Flash Organizations: Crowdsourcing Complex Work By Structuring Crowds As Organizations

Melissa A. Valentine, Daniela Retelny,
Alexandra To, Negar Rahmati, Tulsee Doshi, Michael S. Bernstein
Stanford University
flashorgs@cs.stanford.edu

ABSTRACT

This paper introduces *flash organizations*: crowds structured like organizations to achieve complex and open-ended goals. Microtask workflows, the dominant crowdsourcing structures today, only enable goals that are so simple and modular that their path can be entirely pre-defined. We present a system that organizes crowd workers into computationally-represented structures inspired by those used in organizations — roles, teams, and hierarchies — which support emergent and adaptive coordination toward open-ended goals. Our system introduces two technical contributions: 1) encoding the crowd’s division of labor into de-individualized roles, much as movie crews or disaster response teams use roles to support coordination between on-demand workers who have not worked



When new tools reopen old problems

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcai@google.com

Meredith Ringel Morris
Google DeepMind
Seattle, WA, USA
merrie@google.com

Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu



When you see a new north star

1 [cs.HC] 26 Jul 2023

Embedding Democratic Values into Social Media AIs via Societal Objective Functions

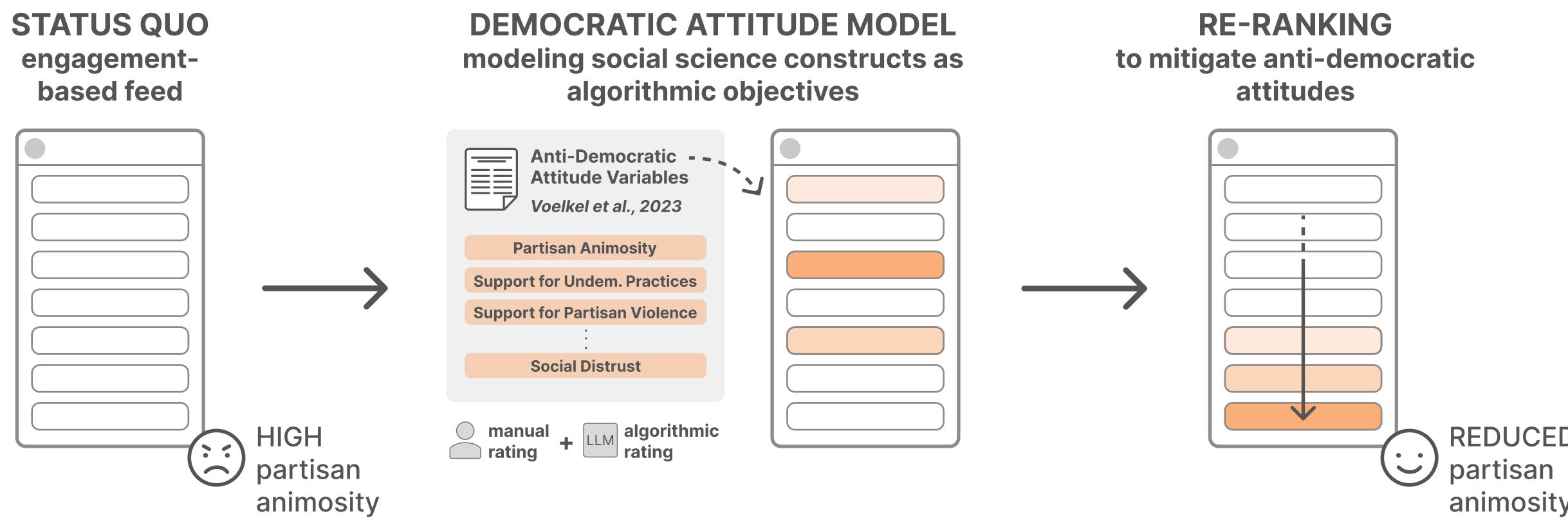
CHENYAN JIA*, Stanford University, USA

MICHELLE S. LAM*, Stanford University, USA

MINH CHAU MAI, Stanford University, USA

JEFFREY T. HANCOCK, Stanford University, USA

MICHAEL S. BERNSTEIN, Stanford University, USA



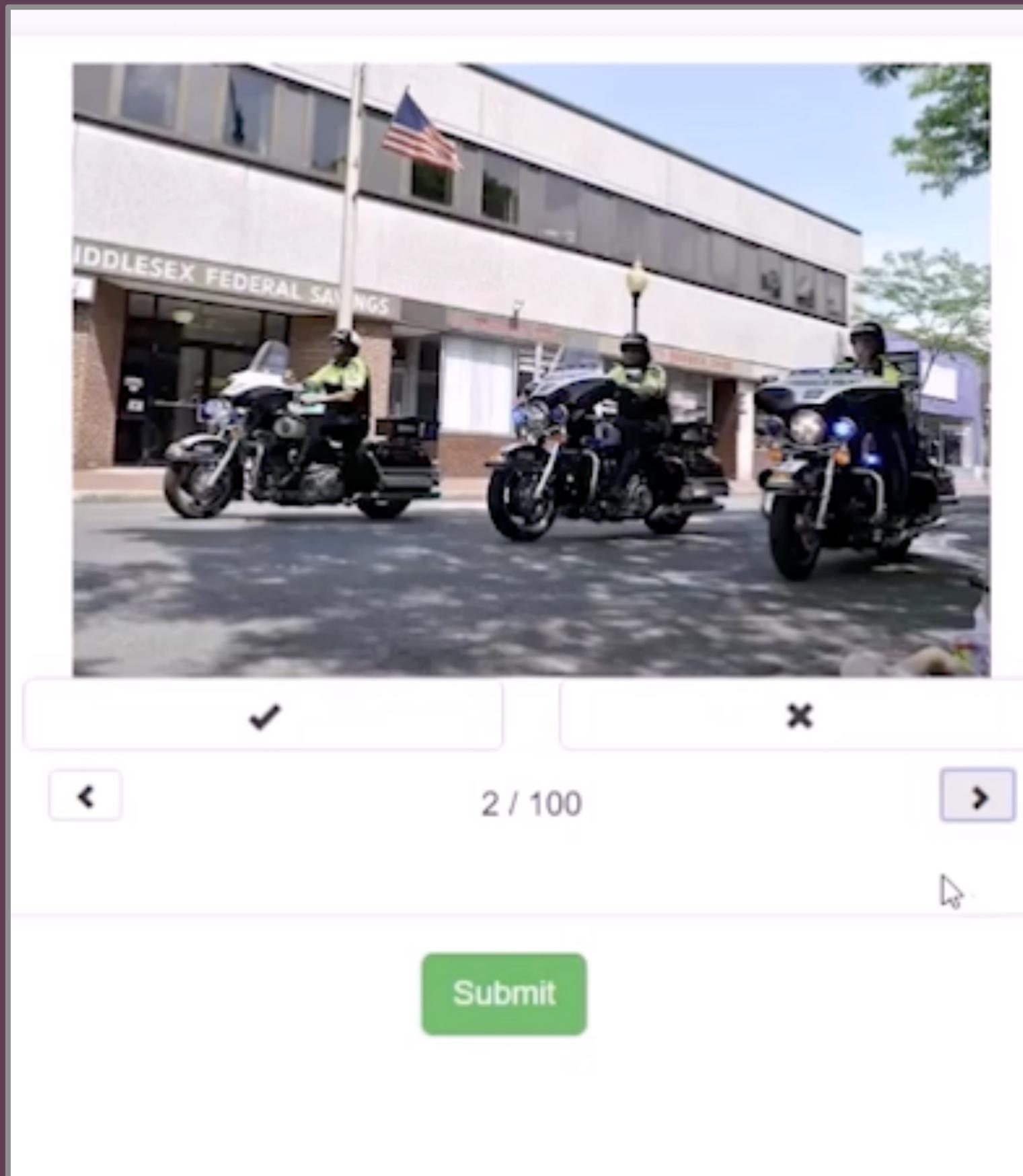
When you see a new north star

Searching for Computer Vision North Stars

Li Fei-Fei & Ranjay Krishna

Computer vision is one of the most fundamental areas of artificial intelligence research. It has contributed to the tremendous progress in the recent deep learning revolution in AI. In this essay, we provide a perspective of the recent evolution of object recognition in computer vision, a flagship research topic that led to the breakthrough data set of ImageNet and its ensuing algorithm developments. We argue that much of this progress is rooted in the pursuit of research “north stars,” wherein researchers focus on critical problems of a scientific discipline that can galvanize major efforts

Playing a hunch: “Hey, would it be possible to...”



Pulling the thread on a weird result

Jury Learning: Integrating Dissenting Voices into Machine Learning Models

Mitchell L. Gordon
Stanford University
Stanford, USA
mgord@cs.stanford.edu

Kayur Patel
Apple Inc.
Seattle, USA
kayur@apple.com

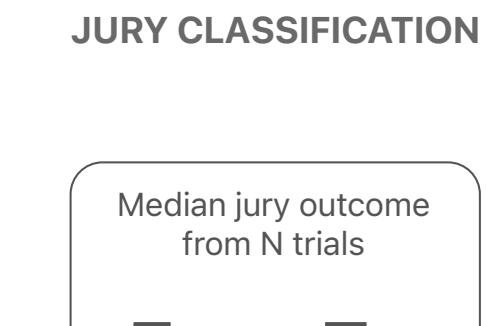
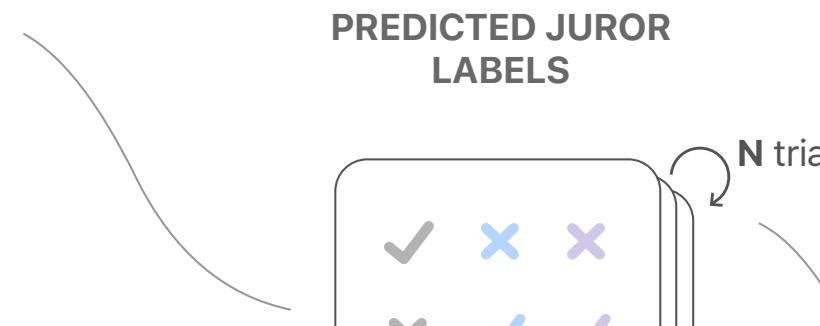
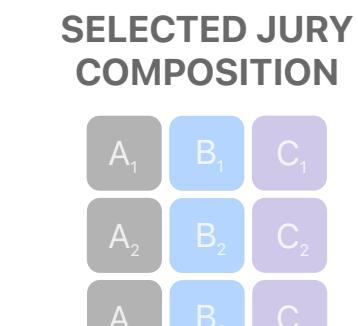
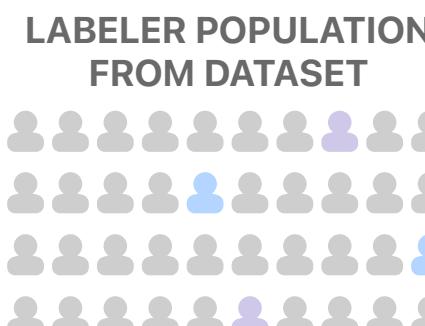
Michelle S. Lam
Stanford University
Stanford, USA
mlam4@stanford.edu

Jeffrey T. Hancock
Stanford University
Stanford, USA
hancockj@stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Tatsunori Hashimoto
Stanford University
Stanford, USA
tatsu@cs.stanford.edu



Which approach do I apply?

This is a skill you develop through mentorship — it's highly contingent, and depends on the problem and solution space that you're navigating.

My suggestion: try on different hats around the problems you're interested in, and see what works.

One final note:

people >> projects

Writing a Paper & Picking Projects