

# Writing a paper & Picking Projects

CS 197 | Stanford University | Kanishk Gandhi  
[cs197.stanford.edu](http://cs197.stanford.edu)

# AI AND HUMANS



# Writing a paper & Picking Projects

CS 197 | Stanford University | Kanishk Gandhi  
[cs197.stanford.edu](http://cs197.stanford.edu)

# 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

Zhen S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Ré, Li Fei-Fei
Stanford University

{zchen,sc,paroma,ranjaykrishna,mb,bernstein,feifeili}@stanford.edu

Abstract

Knowledge bases such as Visual Genome power applications in computer vision, including visual question answering and captioning, but suffer from sparse, noisy relationships. All scene graph models require a training on a small set of visual relationships thousands of training labels each. Hiring human annotators is expensive, and using textual knowledge bases as methods are incompatible with visual data. We introduce a semi-supervised method that establishes relationships labels to a large number of unlabeled images using few labeled examples. We analyze existing methods to see why they are not able to generalize to user generated heuristics, whose out-of-distribution nature is captured by our model. We use as few as 10 labeled examples per relationship to generate enough training labels to train a model on a existing state-of-the-art scene graph model. We show that our method outperforms all baseline approaches on scene graph prediction by 5.3% recall@100 F1 score. In our limited label setting, we define a heuristic for relationships that serves as an indicator for conditions under which our method outperforms transfer learning, the de-facto approach for limited labels.

Introduction

Effort to formalize a structured representation for Visual Genome [27] defined scene graphs, a formalism for representing relationships across images. For example, eat usually consists of one object consuming another object smaller than itself, whereas recall often consists of common objects: phone, laptop or car (see Figure 3). These rules are often learned from a large set of visual question answering [24], relationship [26] and image generation [23]. However, scene graph models ignore more than 98% of top relationships that are not useful for training (see Figure 2) and instead focus on modeling the

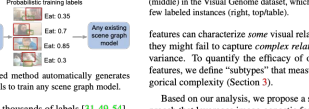


Figure 2. Visual Relationships have a long tail (left) of infrequent relationships. Current models [49, 54] only focus on the top 50 relationships (middle) in the Visual Genome dataset, which have all thousands of labeled instances. This ignores more than 98% of the relationships with few labeled instances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

features can characterize some visual relationships very well, they might fail to capture complex relationships with high variance. To quantify the efficacy of our image-agnostic features, we define "subtypes" that measure spatial and categorical complexity (Section 3). Based on our analysis, we propose a semi-supervised approach that leverages image-agnostic features to label missing relationships using as few as 10 labeled instances of each relationship. We learn simple heuristics over these features and apply them to unlabeled images using a generative model [39, 46]. We evaluate our model's labeling efficacy using the frequently-used VRD dataset [31] and find that it achieves an F1 score of 57.06, which is 11.84 point higher than within 8.65 recall@100 of the same model on like label propagation [57]. To demonstrate the utility of our generated labels, we train a state-of-the-art scene graph model [54] (see Figure 6) and modify its loss function to support 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. We find that our approach improves by 5.16% recall@100 for predicate classification, especially for relationships with high complexity, as it generalizes well to unlabeled subtypes.

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 relationships and using it as a strong prior for learning (Section 5.3). Our semi-supervised model's improvements over transfer learning (Section 5.3).

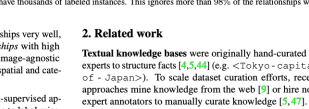


Figure 3. Visual Relationships, such as fly\_eat and sit can be characterized effectively by their categorical (s) and/or refer to subject and object, respectively or spatial features. Some relationships like fly\_eat rely heavily only on a few features — like eat are often seen high up in the sky.

Textual knowledge bases were originally hand-curated by experts to structure facts [8, 45, 44] (e.g. < Tokyo > 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 settings, we use textual knowledge bases to extract and exploit patterns in unlabeled sentences [2, 21, 35, 37, 47]. Unfortunately, such approaches cannot be directly applied to visual relationships, textual relations can often be captured by labels, how well do our visual relationships are often local to an image.

Visual relationships have been studied as spatial priors [14, 16], co-occurrences [51], language statistics [23, 31, 53], and with syntactic models [23]. Scene graph prediction models like label propagation [57]. To demonstrate the utility of our generated labels, we train a state-of-the-art scene graph model [54] (see Figure 6) and modify its loss function to support 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. We find that our approach improves by 5.16% recall@100 for predicate classification, especially for relationships with high complexity, as it generalizes well to unlabeled subtypes.

The de-facto solution for limited label distributions is transfer learning (see Figure 3), which requires that the source domain used by 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 (e.g. fly\_eat, sit), and the target domain is a set of limited 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 the source domain, but instead uses a small labeled set to annotate the unlabeled set of images D0 without any labeled relationships.

To address the issue of gathering enough training labels, we propose a novel neural network architecture that emerged as a popular prior in this approach. This approach allows model imperceptible changes in order to assign training labels to unlabeled data. Imperceptible labeling sources can come from crowd-sourcing [10], local distribution heuristics [8, 43], multi-task learning [22, 40], and distant su-

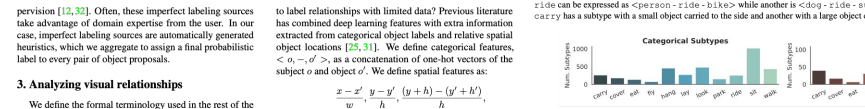


Figure 4. We define the number of subtypes of a relationship as a measure of its complexity. Subtypes can be categorized as expressed as (s, p, o) < D0 > < D1 > while another is (s, o) < D0 > < D1 > < D2 >. Subtype carry has a subtype with a small object carried to the side and another object carried overhead.

perision [12, 23]. Often, these imperfect labeling sources take advantage of domain expertise from the user. In our case, imperfect labels sources are automatically generated heuristics, which we aggregate to assign a final probabilistic label to every pair of object proposals.

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. 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) what labels, how well do our visual 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 category labels to the larger, unlabeled set.

3.1. Terminology

A scene graph is a multi-graph G that consists of objects as nodes and relationships r as edges. Each object o = {o, c} consists of a bounding box b, and its category c ∈ C where C is the set of all possible object categories (e.g. dog, frisbee). Relationships are denoted <subject, predicate, object> or <o, p, o' >, p ∈ P is a predicate, such as ride and eat. We assume that we have a small labeled set {{o, p, o'} ∈ D0} of annotated relationships for each predicate p. Usually, these datasets are on the order of a few examples or fewer. For our semi-supervised method, we also assume that there exists a large set of images D0 without any labeled relationships.

3.2. Defining image-agnostic features

It has become common in computer vision to utilize pre-trained convolutional neural networks to extract features from images. For our scene graph prediction model, we use a set of image embeddings that represent objects and visual relationships [31, 49, 50]. Models trained with these features have proven robust in the presence of enough training data, but tend to overfit when trained on limited data (Section 5). Consequently, an open question arises that other features can we utilize

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

s = x' - y' / (x - y) + (y - h) - (y' - h')

where b = {y, x, w, h} and b' = {y', x', w', h'} 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 spatial and categorical features in Figure 3. Relationships like fly\_eat place high importance on the difference in y-coordinate between the subject and object, capturing a characteristic spatial pattern. Look, on the other hand, depends on the category of the objects (e.g. phone, laptop, window) and not on any spatial orientations.

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. person, eat, sit, bike, or dog can ride a surfboard). To systematically define a notion of complexity, we identify subtypes for each visual relationship. Each subtype captures one way that a relationship manifests itself in the data. For example, a ride contains one categorical subtype with <person, ride> and another with <dog, ride> surfboard. Similarly, a person might carry an object in different related contexts (e.g. carry a bag, carry a suitcase). 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] on the subject and object features on all the

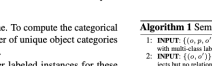


Figure 5. A subset of visual relationships with different levels of complexity is defined by spatial and categorical features. We show how this measure is a good indicator of our semi-supervised method's effectiveness compared to baseline

Algorithm 1 Semi-supervised Ato

- 1: INPUT: {{o, p, o'} ∈ D0}, D1, D2, P — A small labeled set
2: INPUT: G0 — A generative model for unlabeled data
3: INPUT: G1 — A function that extracts features from unlabeled data
4: INPUT: G2 — A heuristic model for unlabeled data
5: INPUT: G3 — A generative model for unlabeled data
6: INPUT: train — Function used to train a model
7: INPUT: eval — Function used to evaluate a model
8: X0 ← G0({o, p, o'} ∈ D0), D0, P
9: Assign labels to o, o' ∈ D0: A ← D0, A' ← D0
10: Learn generative model G0 and assign probabilities to unlabeled data
11: Train scene graph model SGR ← train(G0)
12: OUTPUT: SGR

from the object proposals extracted from the detector [19] on unlabeled D0, (2) on unlabeled D1, and (3) on unlabeled D2, using the image-agnostic features and the factor-graph based generative model using probabilistic labels to the unlabeled data.

These probabilistic labels, along with any scene graph prediction model, are used in Algorithm 1 and show the end-to-end Feature extraction: Our approach uses features defined in Section 3, which we use to train a scene graph prediction model. 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 D0 of unlabeled images in three steps: (1) to extract image-agnostic features from the objects in labeled D0, (2)

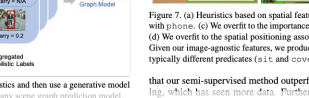


Figure 7. (a) Heuristics based on spatial features help find <man\_fly\_kites>. (b) Our model finds that look is highly correlated with phone. (c) We overfit to the importance of chair as a categorical feature for sit, and fail to identify hang as the correct relationship. (d) We overfit to the spatial positioning associated with sit, where objects are typically longer and directly underneath the subject. (e) Even our image-agnostic features, we produce a reasonable label for chair\_carry, which is incorrect, as two typically different predicates (sit and carry) share a semantic meaning in the context of <classes> r < face2 >

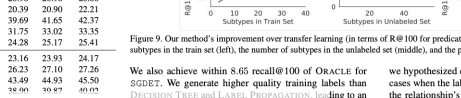
Recall@K as a standard measure for scene graph prediction [11]. Table with columns: Model, R@20, R@50, R@100. Rows include Baseline, FBEG, TRANSFER LEARNING, DISCRETE TRIPLES, LABEL PROPAGATION [57], OURS (DEEP), OURS (SPAT), OURS (MAGNITUDE), OURS (CATEG + SPAT + DEEP), OURS (CATEG + SPAT + WORDVEG), OURS (CATEG + SPAT).

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. Typical past scene graph prediction models have used the Visual Genome to study visual relationships. Unfortunately, these train sets are highly noisy on and off as seen in Figure 8. We compare our model to the state-of-the-art scene graph models, which use the Visual Genome which annotates visual relationships with human labels. Our model is trained on D0, this is a set of bounding boxes, which are used to generate a decision tree over the image-agnostic features, learns from labeled examples in D0, and assigns labels to D1. LABEL PROPAGATION [57] employs a widely-used semi-supervised method and considers the distribution of image-agnostic features in D0 before propagating labels from D0 to D1.

We compare to more frequent baselines (FBEG) using the object counts as priors to make relationship predictions and FBEG+OVERLAP increments such counts only if the object counts are different in the y-coordinate. In our spatial features, (CATEG + SPAT + DEEP) combines all three, and OURS (CATEG + SPAT + WORDVEG) includes word vectors as richer representations of the cate-

Table 2. Results for scene graph prediction tasks with n = 10 labeled examples per predicate, reported as recall@K. A state-of-the-art scene graph trained on labels from our method outperforms those trained with labels generated by other baselines, like transfer learning. Table with columns: Model, R@20, R@50, R@100. Rows include Baseline, FBEG, TRANSFER LEARNING, DISCRETE TRIPLES, LABEL PROPAGATION [57], OURS (DEEP), OURS (SPAT), OURS (MAGNITUDE), OURS (CATEG + SPAT + DEEP), OURS (CATEG + SPAT + WORDVEG), OURS (CATEG + SPAT), ORACLE (HARD = 108%).

gories and predicate labels, and (iii) predicate classification (FBEG), which exploits ground truth bounding boxes and object categories to predict predicate labels. We refer the reader to the paper that introduced these tasks for more details [11]. Finally, we explore how relationship complexity, measured using our definition of subtypes, is correlated with our model's performance relative to transfer learning (Section 5.3).



gories and predicate labels, and (iii) predicate classification (FBEG), which exploits ground truth bounding boxes and object categories to predict predicate labels. We refer the reader to the paper that introduced these tasks for more details [11]. Finally, we explore how relationship complexity, measured using our definition of subtypes, is correlated with our model's performance relative to transfer learning (Section 5.3).

We compare to more frequent baselines (FBEG) using the object counts as priors to make relationship predictions and FBEG+OVERLAP increments such counts only if the object counts are different in the y-coordinate. In our spatial features, (CATEG + SPAT + DEEP) combines all three, and OURS (CATEG + SPAT + WORDVEG) includes word vectors as richer representations of the cate-

a relationship (e.g. carry), we use image-agnostic features to automatically create heuristics and then use a generative model to propagate probabilistic labels to a large unlabeled set of images. These heuristics are threshold-based conditions that are learned from the decision tree. To limit the number of these heuristics and thereby prevent overfitting, we use a decision trees [38] with different restrictions on each feature set to produce J different decision trees and predict labels for the unlabeled data by combining each A ∈ R^{|D0|} × |D0| matrix of predicted labels relationships. However, we only use these heuristics when they have a confidence above the decision threshold. To limit the number of labels with confidence less than a threshold (we chosen to be 2x) random to an abstain, or a abstain, or a abstain. An example of a heuristic is shown in Figure 8. The subject above the object, it assigns a label for the predicate.

We validate our approach for labeling missing relationships using only n = 10 labeled examples by evaluating our probabilistic labels from our semi-supervised approach using the Visual Genome dataset. We use the fully-annotated Visual Genome dataset for training. We use the Visual Genome to study visual relationships. Unfortunately, these train sets are highly noisy on and off as seen in Figure 8. We compare our model to the state-of-the-art scene graph models, which use the Visual Genome which annotates visual relationships with human labels. Our model is trained on D0, this is a set of bounding boxes, which are used to generate a decision tree over the image-agnostic features, learns from labeled examples in D0, and assigns labels to D1. LABEL PROPAGATION [57] employs a widely-used semi-supervised method and considers the distribution of image-agnostic features in D0 before propagating labels from D0 to D1.

gories and predicate labels, and (iii) predicate classification (FBEG), which exploits ground truth bounding boxes and object categories to predict predicate labels. We refer the reader to the paper that introduced these tasks for more details [11]. Finally, we explore how relationship complexity, measured using our definition of subtypes, is correlated with our model's performance relative to transfer learning (Section 5.3).

We compare to more frequent baselines (FBEG) using the object counts as priors to make relationship predictions and FBEG+OVERLAP increments such counts only if the object counts are different in the y-coordinate. In our spatial features, (CATEG + SPAT + DEEP) combines all three, and OURS (CATEG + SPAT + WORDVEG) includes word vectors as richer representations of the cate-

We also achieve within 8.65 recall@100 of ORACLE for SGRDT. We generate higher quality training labels than other methods, which have been used to train any state-of-the-art scene graph prediction model. We assume that in the long-tail of infrequent relationships, we have a small labeled set {{o, p, o'} ∈ D0} of annotated relationships for each predicate (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. Our approach assigns probabilistic labels to a set D0 of unlabeled images in three steps: (1) to extract image-agnostic features from the objects in labeled D0, (2)

we hypothesized earlier, TRANSFER cases when the labeled set only captures a fraction of the scene graph's subtypes. This explains how OURS (CATEG + SPAT) gives a small portion of labeled subtypes. We introduce the first method to complete visual relationships. We define categorical features that use image-agnostic features in a generative model that uses the probabilistic labels to unlabeled images performs best in F1 score where relationships in the complete VRD dataset. We use to train scene graph prediction models using probabilistic labels. We outperforms transfer learning and come close to oracle performance. We trained on a fraction of labeled data. We compare to the state-of-the-art scene graph prediction models and show that it is a strong indicator of our method performs compared to such b



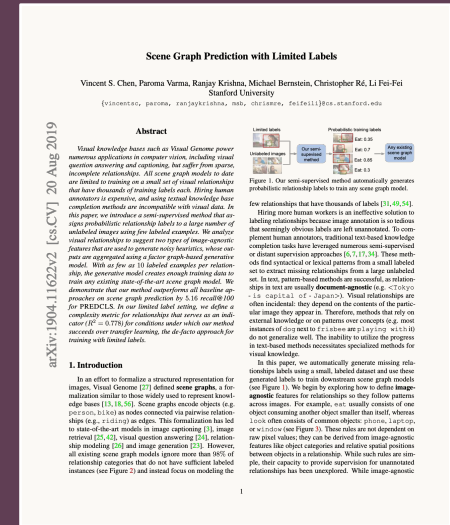
# The common misunderstanding

OK, time to write.

work  
work  
work  
coffee  
work  
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.



...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  $\neq$  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, let the researcher cook:

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



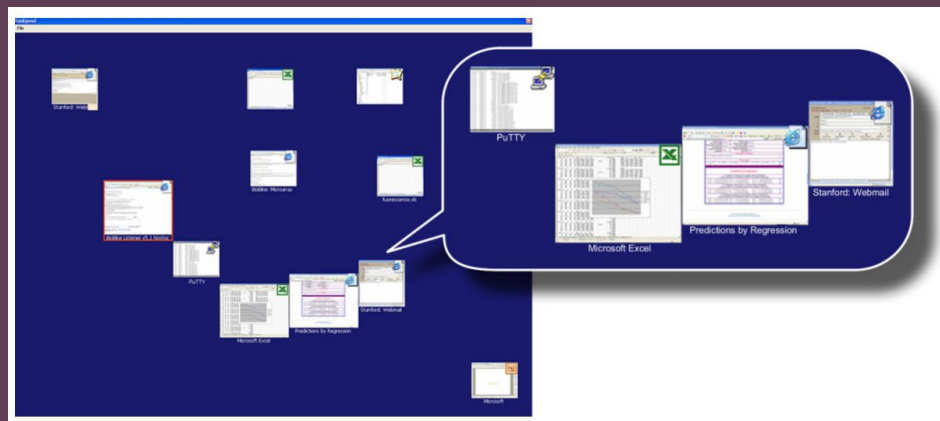
# Riffing is often a good starting point for a first independent project

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

# Even for experienced researchers

Lots of work on  
task-centric workspaces

MSB: "But tasks can have fuzzy  
boundaries!"



MSB (Michael Bernstein)

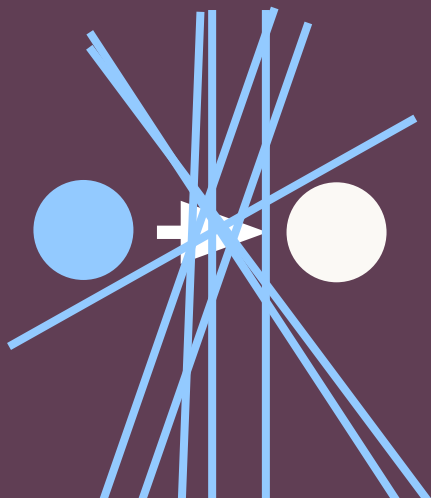


# What are the risks here?

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

A 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

Is this a big rock?

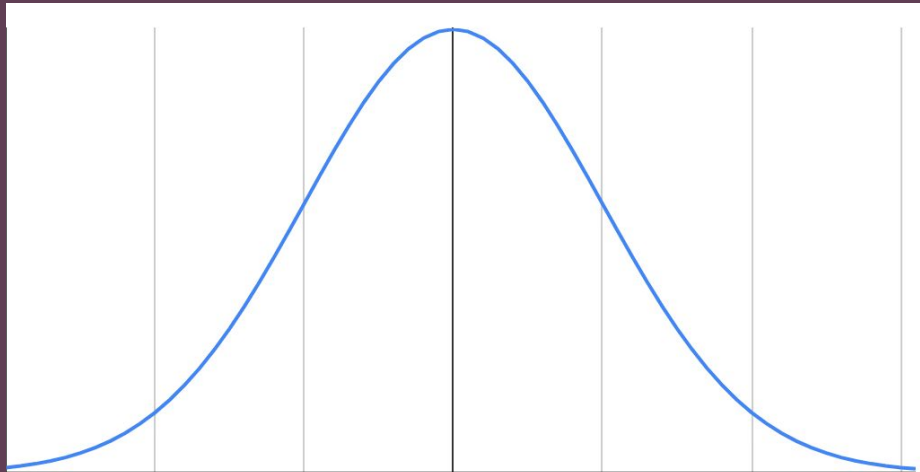
Do I have an angle on it?



“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.

---

## Understanding Social Reasoning in Language Models with Language Models

---

Kanishk Gandhi \* J.-Philipp Fränken \* Tobias Gerstenberg Noah D. Goodman  
Stanford University  
{kanishk.gandhi, jphilipp}@stanford.edu

### Abstract

As Large Language Models (LLMs) become increasingly integrated into our everyday lives, understanding their ability to comprehend human mental states becomes critical for ensuring effective interactions. However, despite the recent attempts to assess the Theory-of-Mind (ToM) reasoning capabilities of LLMs, the degree to which these models can align with human ToM remains a nuanced topic of exploration. This is primarily due to two distinct challenges: (1) the presence of inconsistent results from previous evaluations, and (2) concerns surrounding the validity of existing evaluation methodologies. To address these challenges, we present a novel framework for procedurally generating evaluations *with* LLMs by populating causal templates. Using our framework, we create a new social reasoning benchmark (**BigToM**) for LLMs which consists of 25 controls and 5,000

# 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  
плианг@cs.stanford.edu

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

[cs.HC] 6 Aug 2023



# When you see a new north star

---

## Social Contract AI: Aligning AI Assistants with Implicit Group Norms

---

Jan-Philipp Fränken, Sam Kwok<sup>†</sup>, Peixuan Ye<sup>†</sup>, Kanishk Gandhi

Dilip Arumugam, Jared Moore, Alex Tamkin

Tobias Gerstenberg, Noah D. Goodman

Stanford University  
jphilipp@stanford.edu

### Abstract

We explore the idea of aligning an AI assistant by inverting a model of users' (unknown) preferences from observed interactions. To validate our proposal, we run proof-of-concept simulations in the economic *ultimatum game*, formalizing user preferences as policies that guide the actions of simulated players. We find that the AI assistant accurately *aligns* its behavior to match standard policies from the economic literature (e.g., selfish, altruistic). However, the assistant's learned policies lack robustness and exhibit limited *generalization* in an out-of-distribution setting when confronted with a currency (e.g., grams of medicine) that was not included in the assistant's training distribution. Additionally, we find that when there is *inconsistency* in the relationship between language use and an unknown policy (e.g., an altruistic policy combined with rude language), the assistant's

# 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 and groundbreaking progress. Following the success of ImageNet and object recognition, we observe a number of exciting areas of research and a growing list of north star problems to tackle. This essay recounts the brief history of ImageNet, its related work, and the follow-up progress. The goal is to inspire more north star work to advance the field, and AI at large.*

# 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

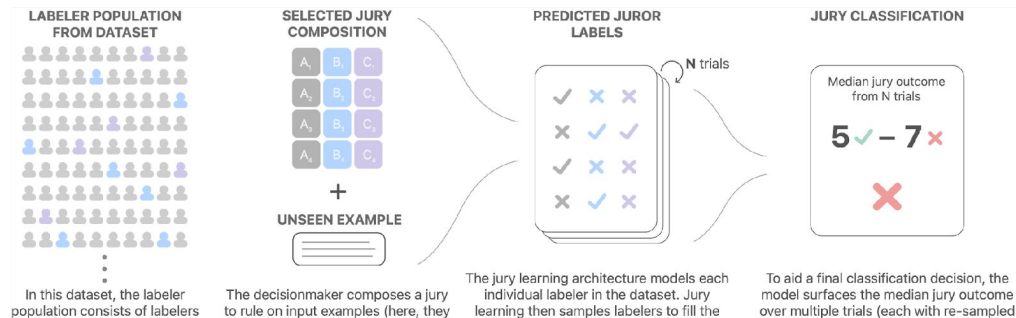
Michelle S. Lam  
Stanford University  
Stanford, USA  
mlam4@stanford.edu

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



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

---

## **Strategic Reasoning with Language Models**

---

**Kanishk Gandhi Dorsa Sadigh Noah D. Goodman**  
Stanford University  
kanishk.gandhi@stanford.edu

### **Abstract**

Strategic reasoning enables agents to cooperate, communicate, and compete with



# Pulling the thread on a weird result

## Eliciting Compatible Demonstrations for Multi-Human Imitation Learning

Kanishk Gandhi, Siddharth Karamcheti, Madeline Liao, Dorsa Sadigh  
Department of Computer Science, Stanford University  
{kanishk.gandhi, skaramcheti, madelineliao, dorsa}@stanford.edu

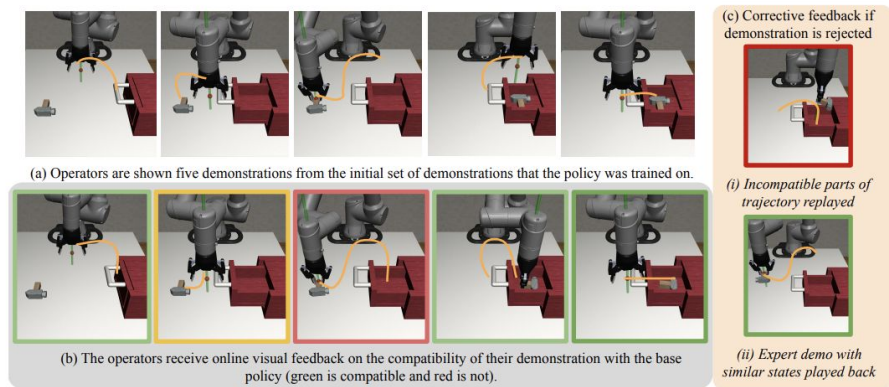


Figure 3: The three phases of our active elicitation interface spanning the initial *prompting* phase (a), the subsequent *demonstration* phase with live feedback (b), and finally, the *feedback* phase (c).

# 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

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