Velocity in Research

CS 197 | Stanford University | Brando Miranda
cs197.stanford.edu

Slides adapted from Kanishk Gandhi & Michael Bernstein

Vectoring in Research

CS 197 | Stanford University | Brando Miranda
cs197.stanford.edu

Slides adapted from Kanishk Gandhi & Michael Bernstein


We Are CSE
Nadia Polikarpova

Administrivia

Finals are scheduled for June 7th (6/7) 7-10pm Friday STLC 115.

Talks for each project: 5 mins + ~2 mins (questions)

What problem are we solving?

"Research is so much slower than industry."

"I feel like we're just not getting anywhere."

"This keeps dragging on and it's not working. I'm losing motivation."

"I missed another submission deadline. I think my advisor is starting to lose faith."

Today’s big idea: velocity

What is research velocity?

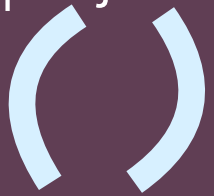
How do we achieve high velocity?

What other signals do people mistake for velocity?

Michael's theory of Researcher success

To be a successful researcher, 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 (often assumption/wrt to main objective/H)



not today!

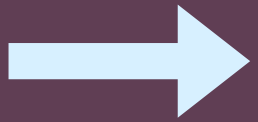
Velocity: rapid reduction of risk in the chosen dimension (you want to learn ASAP – you don't want to "build your life on a lie"! e.g., prototype it vs build expensive infra)

Today!

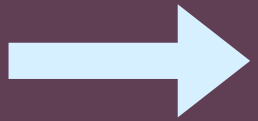
What Is Velocity?

Problematic point of view

"Research is so much slower than industry."

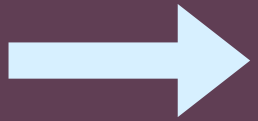


"I feel like we're just not getting anywhere."



We're not making enough progress/suck.

"I missed another submission deadline."



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]

What research is

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

Can be demotivated because it is not linear
Used to classes, put X energy & you get X points back

Need a Mindset shift:
failure = opportunity to learn & improve/grow
uncertainty = opportunity to learn & be curious & investigate
stuck = opportunity to be creative
Did you deliver what you committed, regardless of result?

The Swamp

I have worked on a few projects, and almost every project has a swamp.

The Swamp: challenges that get the project stuck for an extended length of time

E.g.;

Model not performing well

Design not having intended effect

Engineering challenges keep cropping up

& etc

Swamps make progress a poor measure

Swamps can make a project appear to have no or little progress for an extended period of time.

Swamps make progress a bad measure/metric because you might be completing your deliverables (e.g., experiment plots) and learning a lot, even if things are failing!

Progress := "it's working", but you can't control if experiments will work

So progress is a bad measure/metric in high uncertainty projects

Learning = $$ Gold = Failures

However, swamps are when you need to be at your most creative. You need to try many different ideas, and rapidly, to orienteer your way out of a swamp.

The difference between an amazing and a merely good researcher: how effectively and rapidly you explore ways to escape the swamp.

Swamps make progress a poor measure

Recovering quickly & learning better measure/metric!

Yoann Bourgeois

Enter velocity

Drawn from theory and practice of rapid prototyping

Buxton, Sketching User Experiences

Schön, The Reflective Practitioner

Houde and Hill, What Do Prototypes Prototype?

"Enlightened *trial and error* succeeds *over the planning* of the lone genius." - Tom Kelley

Velocity vs. progress

Progress is an absolute delta of your position from the last time we met. How far have you gotten?

Velocity is a measure of the how much you've learned in that time.

If you tried a ton of creative different ideas and they all failed…

that's low progress
but high velocity

Why is velocity a better measure?

Because we are in a high uncertainty landscape, so all you can guarantee is to learn quickly, to learn what is the "correct" thing to be doing & save effort/time
Because failures often mean learning.

Because we likely needed to experience those failures to eventually get to a success: you're learning the landscape.

Because the worst outcome is not failure, but tunneling unproductively - in the "wrong" direction

That's low progress
and low velocity

How do I achieve high velocity?

Restating our goal, precisely

Each week's effort — a draft paper introduction, a user interface, an engineered feature, an evaluation design — is on the path toward understanding the research question.

We have a question to answer this week: Will our hunch work in a simple case? Is assumption X valid? Will this revised model overcome the problematic issue? Can we write a proof for the simple case? We've chosen this week's question that we're trying to answer carefully.

Velocity is the process of answering that question as rapidly as possible.

Choosing this question is the process of vectoring.

Vectoring vs Velocity

Separation of concerns

Vectoring: what is the most risky uncertain idea that can make the project fail?

e.g. Is assumption X valid?

Usually a more abstract idea

Velocity: what exactly should we prototype concretely to learn & derisk quickly

e.g., Build a mock video game in with pen and paper, train small model

Usually concrete, targets the core directly and prototypes periphery

Approach: core vs. periphery

Achieving high velocity means sprinting to answer this week's question, while minimizing all other desiderata for now.

This means being clear with yourself on what you can ignore:

Core: the goal that needs to be achieved in order to answer the question

Periphery: the goals that can be faked, prototyped or assumed, or subsetted, or mocked in, so we can focus on the core question.

Core-periphery mindset

The week's goal is not a demo.

Though this is what is tempting: think, select, and then create.

But this means working on everything both in the core and in the periphery.

The week's goal is instead an answer to a question - learn.

To answer a question, you don't need to address all the issues in the periphery. Just focus on what's in the core.

Make strong assumptions about everything that's in the periphery: use an easy or smaller subset of the data, make simplifying assumptions while working on your proof, ignore other nagging questions for the moment

Be creative & "ruthless" about quickly derisking!

Core-periphery mindset

I'm dedicating a second slide to this concept because it's the key.

Your approach should be, necessarily, incomplete. Do not create a mockup or a scale model. Perfection is your enemy!

Instead, derive everything from your current question:

Will this approach retain all users?
Will this measure correlate with my gut observations?
Will this engineering approach be satisfactory?

Be rapid. Be ruthless. Strip out or fake everything not required to answer the question.

Core-periphery mindset

Seriously: I'm dedicating a third slide to this.

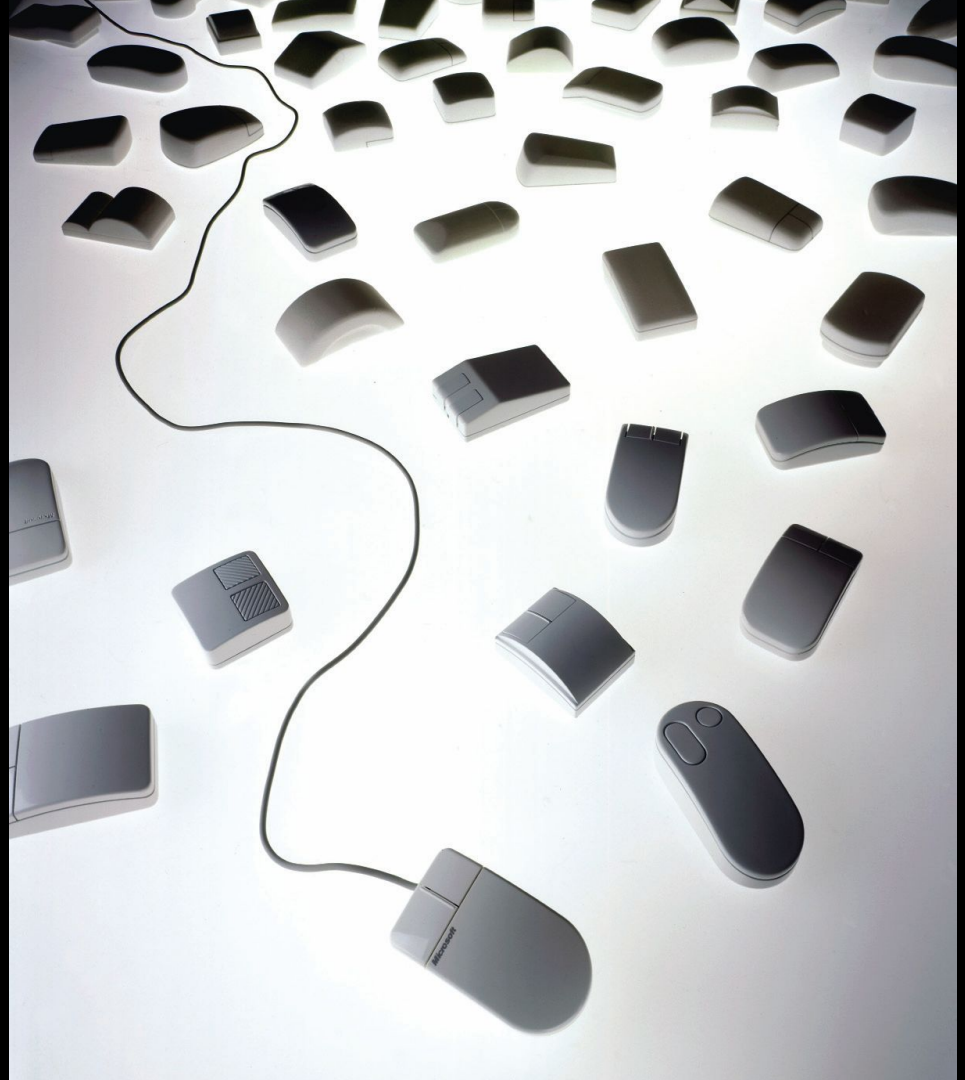
Answer questions, don't engineer. This tends to rankle essentially every facet of your undergraduate training/classes.

Very dangerous to feel you achieved something because you finished coding.

You achieved something if you answered a question, e.g., produced an experiment plot, i.e.: $$ Gold = experiments to learn from

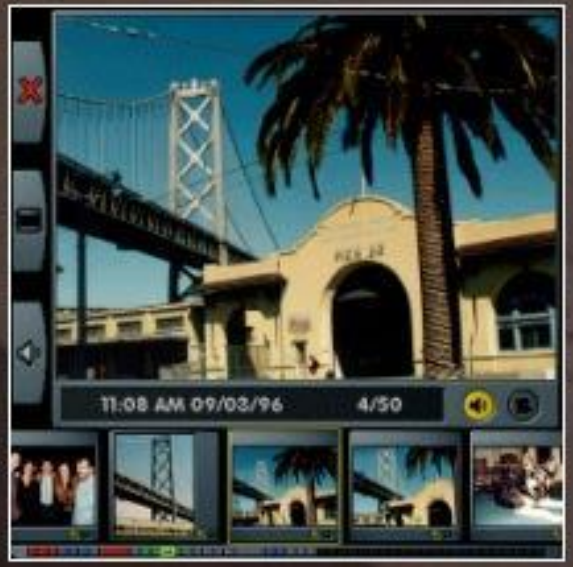
Too often, people pursue perfection in the first pass: perfect drafts, perfectly engineered software, perfect interaction design.

Remember: the goal is to answer the question, not to build that part of your system permanently (yet).



Prototypes of the original Microsoft mouse.

Each one implicitly answering a question.



What question were they asking?

What did they trade off?

All together now

Each week, we engage in vectoring to identify the biggest unanswered question. This should be the focus of your velocity sprint for the week.

To hit high velocity, be strategic about stripping out all other dependencies, faking what you need to, etc., in order to answer the question.

Be prepared to iterate multiple times within the week!

Let’s Try It

Let’s try it out…

Get in groups of 3–4, you’ll have two minutes to discuss each question.

Emergence in LLMs?

Assumption: Everyone thinks emergent capabilities (sharp unpredictable jumps in performance) of LLMs is a fundamental property of scaling AI models

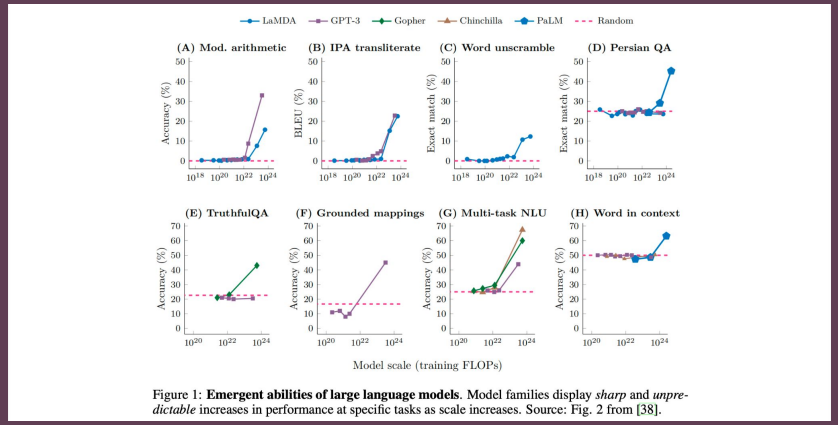


Figure 1: **Emergent abilities of large language models**. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [38].

Hypothesis: authors had a hunch it was mainly due to other factors

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer
Computer Science
Stanford University
rschaef@cs.stanford.edu

Brando Miranda
Computer Science
Stanford University
brando9@cs.stanford.edu

Sanmi Koyejo
Computer Science
Stanford University
sanmi@cs.stanford.edu

Abstract

Recent work claims that large language models display *emergent abilities*: abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due to the researcher's choice of metric rather than due to fundamental changes in models with scale. Specifically, nonlinear or discontinuous metrics produce seemingly emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities; (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on the Beyond the Imitation Game Benchmark (BIG-Bench); and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep network architectures. Via all three analyses, we provide evidence that emergent abilities disappear with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

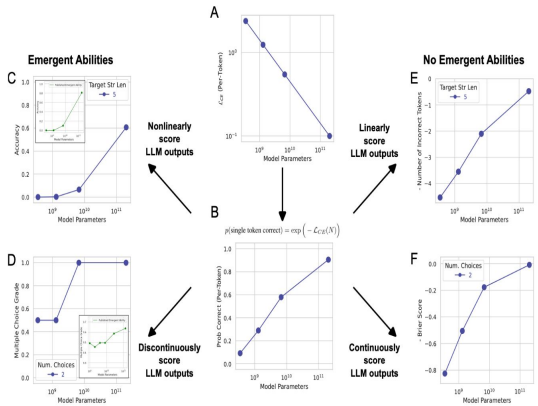


Figure 2: **Emergent abilities of large language models are created by the researcher's chosen metrics, not unpredictable changes in model behavior with scale.** (A) Suppose the per-token

Emergence in LLMs?

Hypothesis: Emergent Capabilities (unpredictable jumps) were possibly due to different factors than fundamental properties of scaling AI models

Vector (highest direction of risk):

Is it due to model scoring metric?

How do we test it as quickly as possible?

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer
Computer Science
Stanford University
rschaef@cs.stanford.edu

Brando Miranda
Computer Science
Stanford University
brando9@cs.stanford.edu

Sanmi Koyejo
Computer Science
Stanford University
sanmi@cs.stanford.edu

Abstract

Recent work claims that large language models display *emergent abilities*: abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due to the researcher's choice of metric rather than due to fundamental changes in models with scale. Specifically, nonlinear or discontinuous metrics produce seemingly emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities; (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on the Beyond the Imitation Game Benchmark (BIG-Bench); and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep network architectures. Via all three analyses, we provide evidence that emergent abilities disappear with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

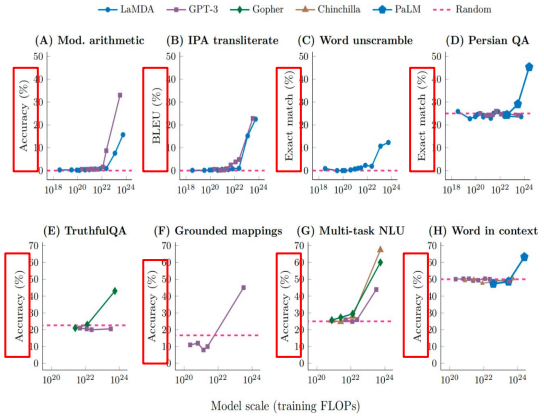


Figure 1: **Emergent abilities of large language models**. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [38].

Emergence in LLMs?

Vector: emergence due model scoring metric?

How do we test it as quickly as possible?

One change; the scoring function

We chose modular arithmetic e.g., quicker and smaller data set to run vs say Multi-task NLU

We could generate data for task, so we were in control of size and speed to learn

Use easily accessible models, GPT3.5 API quicker than using OS LLMs in a cluster e.g., GPU memory issues (engineering)

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer
Computer Science
Stanford University
rschaef@cs.stanford.edu

Brando Miranda
Computer Science
Stanford University
brando9@cs.stanford.edu

Sanmi Koyejo
Computer Science
Stanford University
sanmi@cs.stanford.edu

Abstract

Recent work claims that large language models display *emergent abilities*: abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due to the researcher's choice of metric rather than due to fundamental changes in models with scale. Specifically, nonlinear or discontinuous metrics produce seemingly emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities; (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on the Beyond the Imitation Game Benchmark (BIG-Bench); and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep network architectures. Via all three analyses, we provide evidence that emergent abilities disappear with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

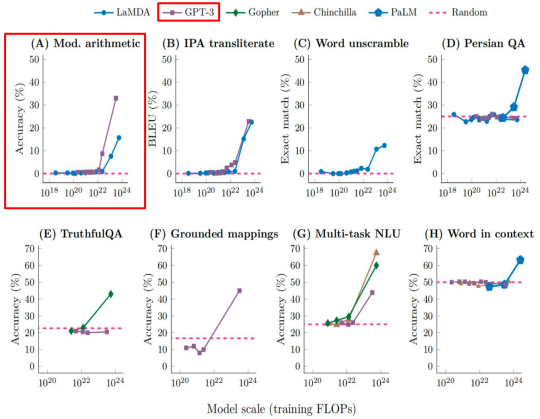


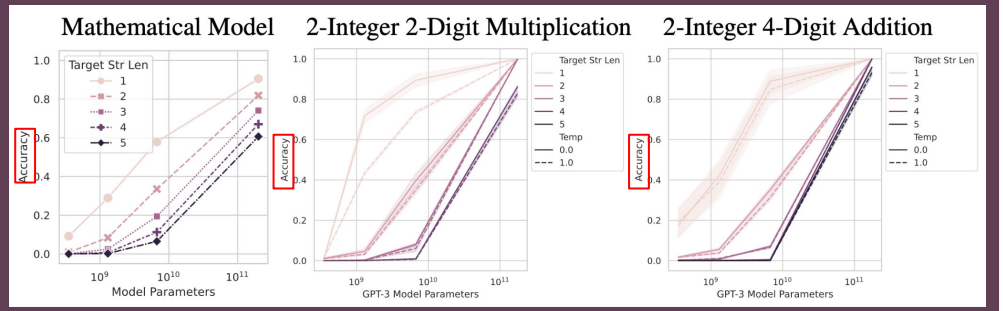
Figure 1: **Emergent abilities of large language models**. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [38].

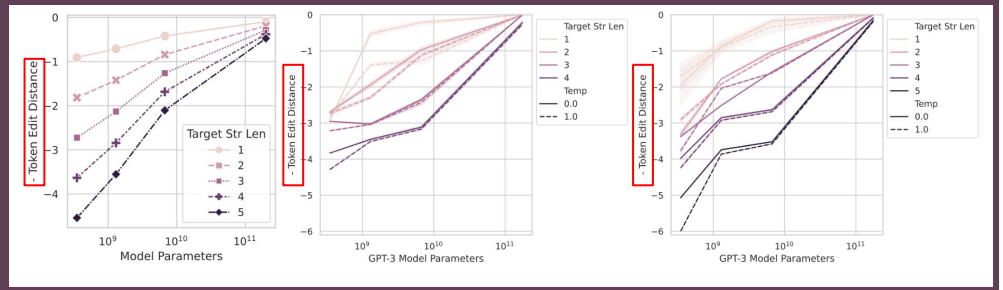
Emergence in LLMs?

Vector: emergence due model scoring metric?

How do we test it as quickly as possible? "$$ cash" = experiment = "learning" (not engineering)

One change, change the scoring function





Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer
Computer Science
Stanford University
rschaef@cs.stanford.edu

Brando Miranda
Computer Science
Stanford University
brando9@cs.stanford.edu

Sanmi Koyejo
Computer Science
Stanford University
sanmi@cs.stanford.edu

Abstract

Recent work claims that large language models display *emergent abilities*: abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due to the researcher's choice of metric rather than due to fundamental changes in models with scale. Specifically, nonlinear or discontinuous metrics produce seemingly emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities; (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on the Beyond the Imitation Game Benchmark (BIG-Bench); and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep network architectures. Via all three analyses, we provide evidence that emergent abilities disappear with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

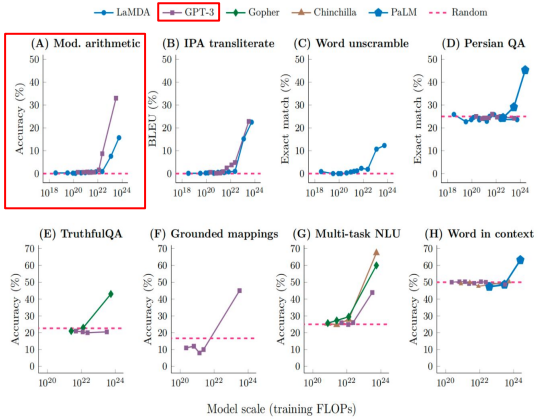


Figure 1: **Emergent abilities of large language models**. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [38].

Emergence in LLMs?

New Vector: emergence due size of test set?
Models that are too small might have 10^-3

chance to get something right but if your test set gas 10 examples, your model will score exactly zero

How do we test it as quickly as possible?

"$$ cash" = experiment = "learning" (not engineering)

Velocity:

Increase test set for modular arithmetic (we have control!)

Persian QA bad idea – you need to hire people that speak Persian!

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer
Computer Science
Stanford University
rschaef@cs.stanford.edu

Brando Miranda
Computer Science
Stanford University
brando9@cs.stanford.edu

Sanmi Koyejo
Computer Science
Stanford University
sanmi@cs.stanford.edu

Abstract

Recent work claims that large language models display *emergent abilities*: abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due to the researcher's choice of metric rather than due to fundamental changes in models with scale. Specifically, nonlinear or discontinuous metrics produce seemingly emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities; (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on the Beyond the Imitation Game Benchmark (BIG-Bench); and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep network architectures. Via all three analyses, we provide evidence that emergent abilities disappear with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

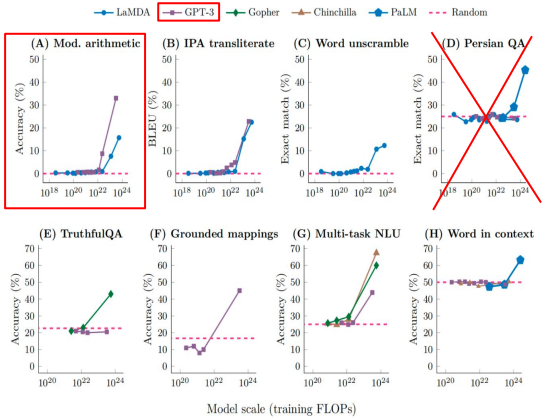


Figure 1: **Emergent abilities of large language models**. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [38].

Emergence in LLMs?

New Vector: emergence due size of test set?

Velocity (prototype, learn quickly):

Increase test set for modular arithmetic (we have control!) & ue GPT3.5

Persian QA bad idea – you need to hire people that speak Persian!

Accuracy, not zero anymore! $$ == Experiments!

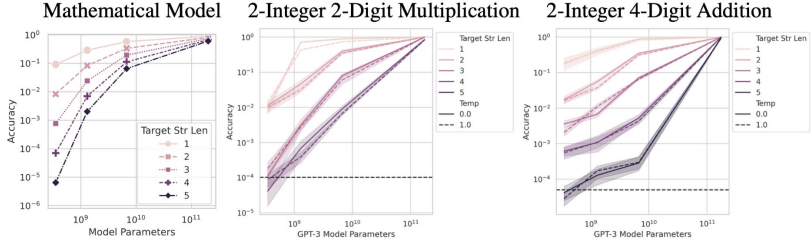


Figure 4: **Claimed emergent abilities evaporate upon using better statistics.** Based on the predictable effect Accuracy has on performance, measuring performance requires high resolution. Generating additional test data increases the resolution and reveals that even on Accuracy, the InstructGPT/GPT-3 family's [4, 27] performance is above chance and improves in a smooth, continuous, predictable manner that qualitatively matches the mathematical model.

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer
Computer Science
Stanford University
rschaef@cs.stanford.edu

Brando Miranda
Computer Science
Stanford University
brando9@cs.stanford.edu

Sanmi Koyejo
Computer Science
Stanford University
sanmi@cs.stanford.edu

Abstract

Recent work claims that large language models display *emergent abilities*: abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due to the researcher's choice of metric rather than due to fundamental changes in models with scale. Specifically, nonlinear or discontinuous metrics produce seemingly emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities; (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on the Beyond the Imitation Game Benchmark (BIG-Bench); and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep network architectures. Via all three analyses, we provide evidence that emergent abilities disappear with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

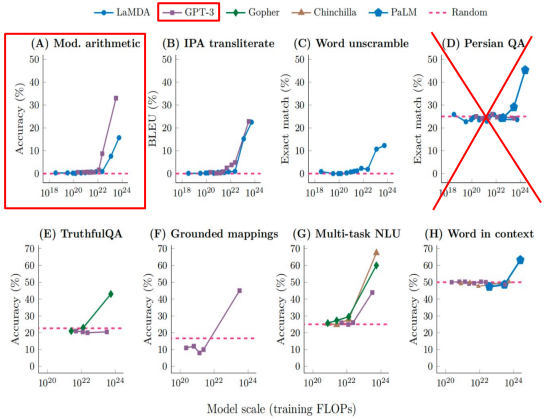


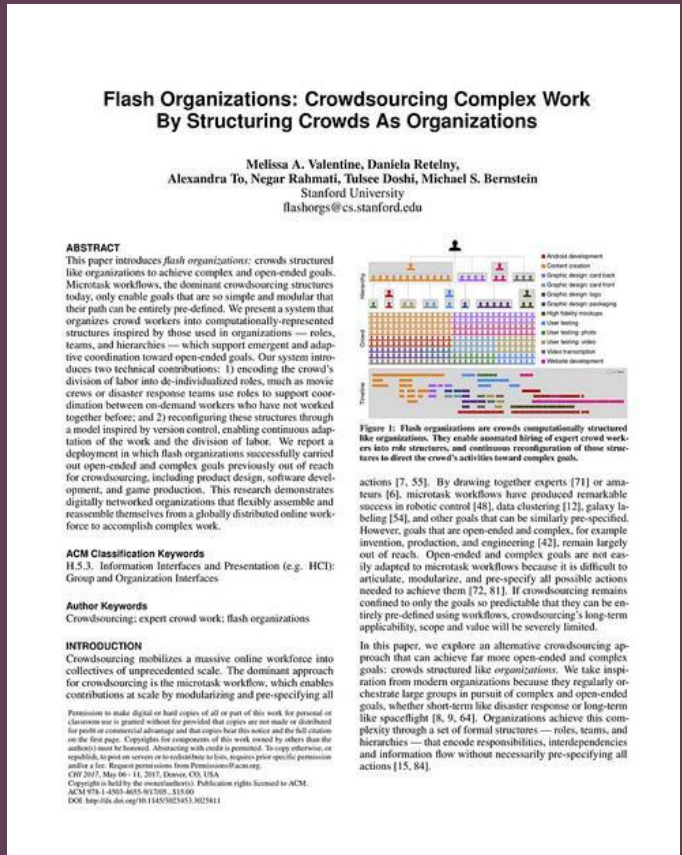
Figure 1: **Emergent abilities of large language models**. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [38].

Social debugging: flash organizations

They had a problem of online workers not being as good as their Upwork profile suggested. They wanted workers who were experts at Angular, Django, UI, UX, marketing, etc, but often in practice they were not as good as they advertised.

Had a hunch that giving workers ~1hr starter tasks would allow us to vet them.

How do you test this hunch?



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 together before; and 2) reconfiguring these structures through a model inspired by version control, enabling continuous adaptation of the work and the division of labor. We report a deployment in which flash organizations successfully carried out open-ended and complex goals previously out of reach for crowdsourcing, including product design, software development, and game production. This research demonstrates digitally networked organizations that flexibly assemble and reassemble themselves from a globally distributed online workforce to accomplish complex work.

ACM Classification Keywords
H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces

Author Keywords
Crowdsourcing; expert crowd work; flash organizations

INTRODUCTION
Crowdsourcing mobilizes a massive online workforce into collectives of unprecedented scale. The dominant approach for crowdsourcing is the microtask workflow, which enables contributions at scale by modularizing and pre-specifying all

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05...$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025811

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

actions [7, 55]. By drawing together experts [71] or amateurs [6], microtask workflows have produced remarkable success in robotic control [48], data clustering [12], galaxy labeling [54], and other goals that can be similarly pre-specified. However, goals that are open-ended and complex, for example invention, production, and engineering [42], remain largely out of reach. Open-ended and complex goals are not easily adapted to microtask workflows because it is difficult to articulate, modularize, and pre-specify all possible actions needed to achieve them [72, 81]. If crowdsourcing remains confined to only the goals so predictable that they can be entirely pre-defined using workflows, crowdsourcing's long-term applicability, scope and value will be severely limited.

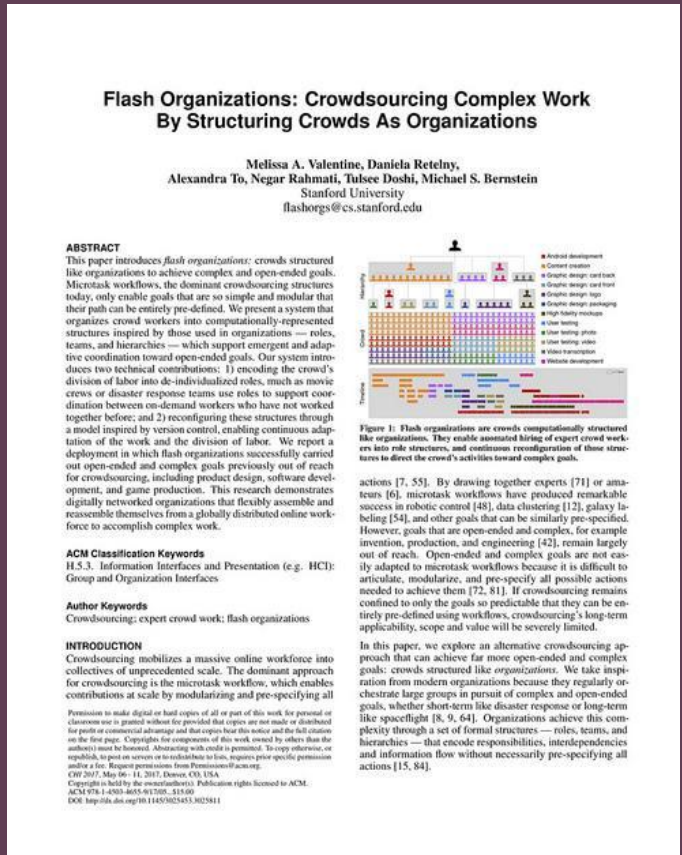
In this paper, we explore an alternative crowdsourcing approach that can achieve far more open-ended and complex goals: crowds structured like *organizations*. We take inspiration from modern organizations because they regularly orchestrate large groups in pursuit of complex and open-ended goals, whether short-term like disaster response or long-term like spaceflight [8, 9, 64]. Organizations achieve this complexity through a set of formal structures — roles, teams, and hierarchies — that encode responsibilities, interdependencies and information flow without necessarily pre-specifying all actions [15, 84].

Social debugging: flash organizations

They picked a small number of domains and manually generated quick test tasks for them. We posted these as jobs, giving a time limit. We manually evaluated the results.

They didn't care about generalizability or software integration.

Later, they asked: could this scale to hundreds of people and tens of domains?



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 together before; and 2) reconfiguring these structures through a model inspired by version control, enabling continuous adaptation of the work and the division of labor. We report a deployment in which flash organizations successfully carried out open-ended and complex goals previously out of reach for crowdsourcing, including product design, software development, and game production. This research demonstrates digitally networked organizations that flexibly assemble and reassemble themselves from a globally distributed online workforce to accomplish complex work.

ACM Classification Keywords
H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces

Author Keywords
Crowdsourcing; expert crowd work; flash organizations

INTRODUCTION
Crowdsourcing mobilizes a massive online workforce into collectives of unprecedented scale. The dominant approach for crowdsourcing is the microtask workflow, which enables contributions at scale by modularizing and pre-specifying all

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05...$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025811

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

actions [7, 55]. By drawing together experts [71] or amateurs [6], microtask workflows have produced remarkable success in robotic control [48], data clustering [12], galaxy labeling [54], and other goals that can be similarly pre-specified. However, goals that are open-ended and complex, for example invention, production, and engineering [42], remain largely out of reach. Open-ended and complex goals are not easily adapted to microtask workflows because it is difficult to articulate, modularize, and pre-specify all possible actions needed to achieve them [72, 81]. If crowdsourcing remains confined to only the goals so predictable that they can be entirely pre-defined using workflows, crowdsourcing's long-term applicability, scope and value will be severely limited.

In this paper, we explore an alternative crowdsourcing approach that can achieve far more open-ended and complex goals: crowds structured like *organizations*. We take inspiration from modern organizations because they regularly orchestrate large groups in pursuit of complex and open-ended goals, whether short-term like disaster response or long-term like spaceflight [8, 9, 64]. Organizations achieve this complexity through a set of formal structures — roles, teams, and hierarchies — that encode responsibilities, interdependencies and information flow without necessarily pre-specifying all actions [15, 84].

Mutual Exclusivity

Children use the mutual exclusivity (ME) bias to help disambiguate how words map to referents, assuming that if an object has one label then it does not need another.

We had a hunch that neural networks won't show this bias.

How do you quickly test this?

Mutual exclusivity as a challenge for deep neural networks

Kanishk Gandhi
New York University
kanishk.gandhi@nyu.edu

Brenden Lake
New York University
Facebook AI Research
brenden@nyu.edu

Abstract

Strong inductive biases allow children to learn in fast and adaptable ways. Children use the mutual exclusivity (ME) bias to help disambiguate how words map to referents, assuming that if an object has one label then it does not need another. In this paper, we investigate whether or not vanilla neural architectures have an ME bias, demonstrating that they lack this learning assumption. Moreover, we show that their inductive biases are poorly matched to lifelong learning formulations of classification and translation. We demonstrate that there is a compelling case for designing task-general neural networks that learn through mutual exclusivity, which remains an open challenge.

1 Introduction

Children are remarkable learners, and thus their inductive biases should interest machine learning researchers. To help learn the meaning of new words efficiently, children use the "mutual exclusivity" (ME) bias – the assumption that once an object has one name, it does not need another [1] (Figure 1). In this paper, we examine whether or not vanilla neural networks demonstrate the mutual exclusivity bias, either as a built-in assumption or as a bias that develops through training. Moreover, we examine common benchmarks in machine translation and object recognition to determine whether or not a maximally efficient learner should use mutual exclusivity.

When children endeavour to learn a new word, they rely on inductive biases to narrow the space of possible meanings. Children learn an average of about 10 new words per day from the age of one until the end of high school [2], a feat that requires managing a tractable set of candidate meanings. A typical word learning scenario has many sources of ambiguity and uncertainty, including ambiguity in the mapping between words and referents. Children hear multiple words and see multiple objects within a single scene, often without clear supervisory signals to indicate which word goes with which object [3].

Figure 1: The mutual exclusivity task used in cognitive development research [1]. Children tend to associate the novel word ("dax") with the novel object (right).

The mutual exclusivity assumption helps to resolve ambiguity in how words map to their referents. Markman and Watchel [1] examined scenarios like Figure 1 that required children to determine the referent of a novel word. For instance, children who know the meaning of "cup" are presented with two objects, one which is familiar (a cup) and another which is novel (an unusual object). Given these two objects, children are asked to "Show me a dax," where "dax" is a novel nonsense word. Markman and Wachtel found that children tend to pick the novel object rather than the familiar one. Although it is possible that the word "dax" could be another word for referring to cups, children predict that the novel word refers to the novel object – demonstrating a "mutual exclusivity" bias that familiar objects do not need another name. This is only

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Mutual Exclusivity

We used a rough simulation!

Map a one-hot to vector to another one-hot vector.

Train a small neural network, ~5 minutes locally.

Next Step: Does this work with more realistic data? Can small variations in training change this?

Mutual exclusivity as a challenge for deep neural networks

Kanishk Gandhi
New York University
kanishk.gandhi@nyu.edu

Brenden Lake
New York University
Facebook AI Research
brenden@nyu.edu

Abstract

Strong inductive biases allow children to learn in fast and adaptable ways. Children use the mutual exclusivity (ME) bias to help disambiguate how words map to referents, assuming that if an object has one label then it does not need another. In this paper, we investigate whether or not vanilla neural architectures have an ME bias, demonstrating that they lack this learning assumption. Moreover, we show that their inductive biases are poorly matched to lifelong learning formulations of classification and translation. We demonstrate that there is a compelling case for designing task-general neural networks that learn through mutual exclusivity, which remains an open challenge.

1 Introduction

Children are remarkable learners, and thus their inductive biases should interest machine learning researchers. To help learn the meaning of new words efficiently, children use the "mutual exclusivity" (ME) bias – the assumption that once an object has one name, it does not need another [1] (Figure 1). In this paper, we examine whether or not vanilla neural networks demonstrate the mutual exclusivity bias, either as a built-in assumption or as a bias that develops through training. Moreover, we examine common benchmarks in machine translation and object recognition to determine whether or not a maximally efficient learner should use mutual exclusivity.

When children endeavour to learn a new word, they rely on inductive biases to narrow the space of possible meanings. Children learn an average of about 10 new words per day from the age of one until the end of high school [2], a feat that requires managing a tractable set of candidate meanings. A typical word learning scenario has many sources of ambiguity and uncertainty, including ambiguity in the mapping between words and referents. Children hear multiple words and see multiple objects within a single scene, often without clear supervisory signals to indicate which word goes with which object [3].

Figure 1: The mutual exclusivity task used in cognitive development research [1]. Children tend to associate the novel word ("dax") with the novel object (right).

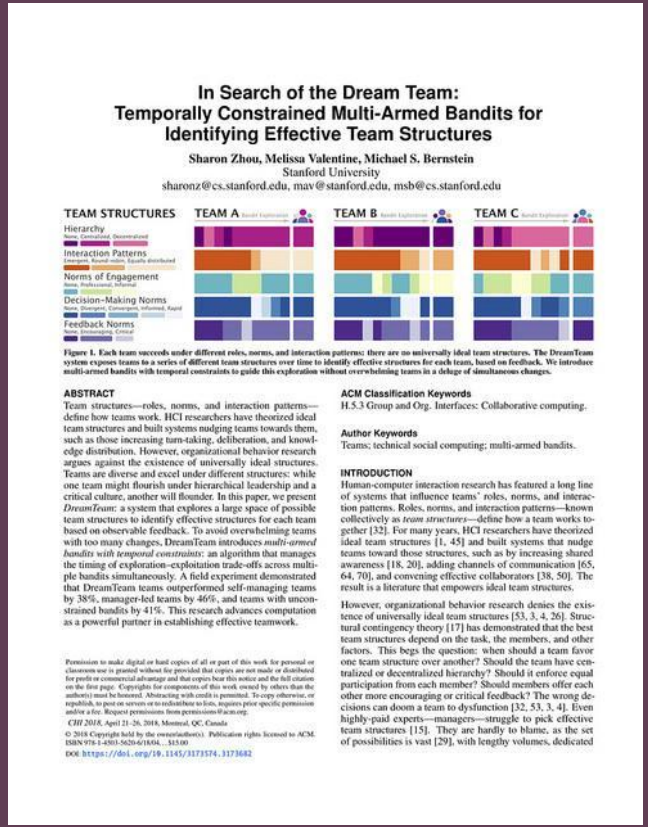
The mutual exclusivity assumption helps to resolve ambiguity in how words map to their referents. Markman and Watchel [1] examined scenarios like Figure 1 that required children to determine the referent of a novel word. For instance, children who know the meaning of "cup" are presented with two objects, one which is familiar (a cup) and another which is novel (an unusual object). Given these two objects, children are asked to "Show me a dax," where "dax" is a novel nonsense word. Markman and Wachtel found that children tend to pick the novel object rather than the familiar one. Although it is possible that the word "dax" could be another word for referring to cups, children predict that the novel word refers to the novel object – demonstrating a "mutual exclusivity" bias that familiar objects do not need another name. This is only

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Engineering: Dream Team

This project used multi-armed bandits to identify over several rounds of interaction whether teams should be flat or hierarchical, supportive or critical, etc. But we didn't know: could these multi-armed bandits actually converge fast enough to be useful?

We had a rough implementation of the multi-armed bandits, but it wasn't production ready for interacting with teams.



In Search of the Dream Team: Temporally Constrained Multi-Armed Bandits for Identifying Effective Team Structures

Sharon Zhou, Melissa Valentine, Michael S. Bernstein
Stanford University
sharonz@cs.stanford.edu, mav@stanford.edu, msb@cs.stanford.edu

Figure 1. Each team succeeds under different roles, norms, and interaction patterns; there are no universally ideal team structures. The DreamTeam system exposes teams to a series of different team structures over time to identify effective structures for each team, based on feedback. We introduce multi-armed bandits with temporal constraints to guide this exploration without overwhelming teams in a deluge of simultaneous changes.

ABSTRACT

Team structures—roles, norms, and interaction patterns—define how teams work. HCI researchers have theorized ideal team structures and built systems nudging teams towards them, such as those increasing turn-taking, deliberation, and knowledge distribution. However, organizational behavior research argues against the existence of universally ideal structures. Teams are diverse and excel under different structures: while one team might flourish under hierarchical leadership and a critical culture, another will flounder. In this paper, we present *DreamTeam*: a system that explores a large space of possible team structures to identify effective structures for each team based on observable feedback. To avoid overwhelming teams with too many changes, DreamTeam introduces *multi-armed bandits with temporal constraints*: an algorithm that manages the timing of exploration–exploitation trade-offs across multiple bandits simultaneously. A field experiment demonstrated that DreamTeam teams outperformed self-managing teams by 38%, manager-led teams by 46%, and teams with unconstrained bandits by 41%. This research advances computation as a powerful partner in establishing effective teamwork.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ISBN 978-1-4503-5620-6/18/04…$15.00
DOI: https://doi.org/10.1145/3173574.3173682

ACM Classification Keywords
H.5.3 Group and Org. Interfaces: Collaborative computing.

Author Keywords
Teams; technical social computing; multi-armed bandits.

INTRODUCTION

Human-computer interaction research has featured a long line of systems that influence teams' roles, norms, and interaction patterns. Roles, norms, and interaction patterns—known collectively as *team structures*—define how a team works together [32]. For many years, HCI researchers have theorized ideal team structures [1, 45] and built systems that nudge teams toward those structures, such as by increasing shared awareness [18, 20], adding channels of communication [65, 64, 70], and convening effective collaborators [38, 50]. The result is a literature that empowers ideal team structures.

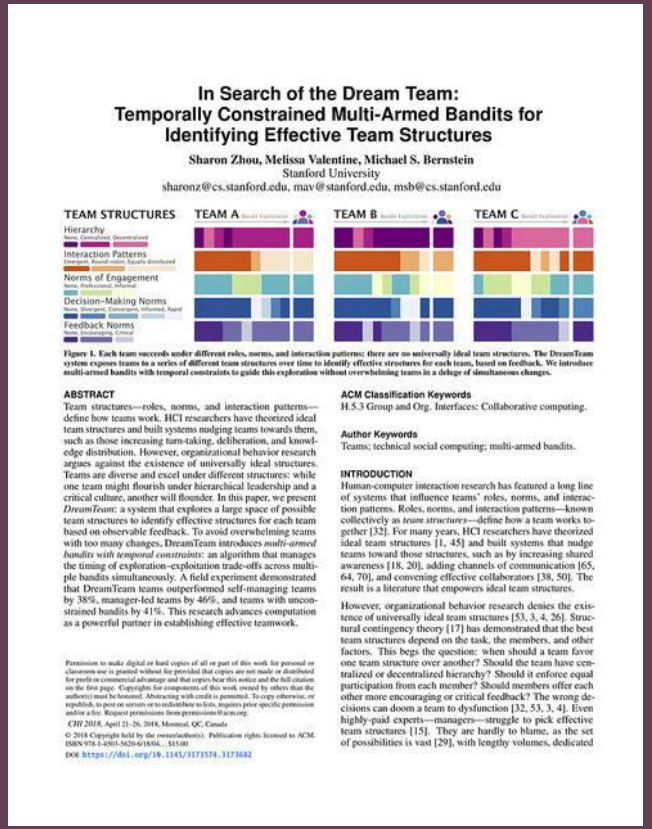
However, organizational behavior research denies the existence of universally ideal team structures [53, 3, 4, 26]. Structural contingency theory [17] has demonstrated that the best team structures depend on the task, the members, and other factors. This begs the question: when should a team favor one team structure over another? Should the team have centralized or decentralized hierarchy? Should it enforce equal participation from each member? Should members offer each other more encouraging or critical feedback? The wrong decisions can doom a team to dysfunction [32, 53, 3, 4]. Even highly-paid experts—managers—struggle to pick effective team structures [15]. They are hardly to blame, as the set of possibilities is vast [29], with lengthy volumes, dedicated

Engineering: Dream Team

We used a rough simulation! Assuming some roughly accurate numbers in how much each team benefited from each bandit setting, we generated teams and simulated the bandits over a few rounds.

The answer: they converged quickly enough that this might work!

(The next step: wizard of oz the interface, so we could test it “for real” without building integrating software.)



In Search of the Dream Team: Temporally Constrained Multi-Armed Bandits for Identifying Effective Team Structures

Sharon Zhou, Melissa Valentine, Michael S. Bernstein
Stanford University
sharonz@cs.stanford.edu, mav@stanford.edu, msb@cs.stanford.edu

Figure 1. Each team succeeds under different roles, norms, and interaction patterns; there are no universally ideal team structures. The DreamTeam system exposes teams to a series of different team structures over time to identify effective structures for each team, based on feedback. We introduce multi-armed bandits with temporal constraints to guide this exploration without overwhelming teams in a deluge of simultaneous changes.

ABSTRACT

Team structures—roles, norms, and interaction patterns—define how teams work. HCI researchers have theorized ideal team structures and built systems nudging teams towards them, such as those increasing turn-taking, deliberation, and knowledge distribution. However, organizational behavior research argues against the existence of universally ideal structures. Teams are diverse and excel under different structures: while one team might flourish under hierarchical leadership and a critical culture, another will flounder. In this paper, we present *DreamTeam*: a system that explores a large space of possible team structures to identify effective structures for each team based on observable feedback. To avoid overwhelming teams with too many changes, DreamTeam introduces *multi-armed bandits with temporal constraints*: an algorithm that manages the timing of exploration–exploitation trade-offs across multiple bandits simultaneously. A field experiment demonstrated that DreamTeam teams outperformed self-managing teams by 38%, manager-led teams by 46%, and teams with unconstrained bandits by 41%. This research advances computation as a powerful partner in establishing effective teamwork.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ISBN 978-1-4503-5620-6/18/04…$15.00
DOI: https://doi.org/10.1145/3173574.3173682

ACM Classification Keywords
H.5.3 Group and Org. Interfaces: Collaborative computing.

Author Keywords
Teams; technical social computing; multi-armed bandits.

INTRODUCTION

Human-computer interaction research has featured a long line of systems that influence teams' roles, norms, and interaction patterns. Roles, norms, and interaction patterns—known collectively as *team structures*—define how a team works together [32]. For many years, HCI researchers have theorized ideal team structures [1, 45] and built systems that nudge teams toward those structures, such as by increasing shared awareness [18, 20], adding channels of communication [65, 64, 70], and convening effective collaborators [38, 50]. The result is a literature that empowers ideal team structures.

However, organizational behavior research denies the existence of universally ideal team structures [53, 3, 4, 26]. Structural contingency theory [17] has demonstrated that the best team structures depend on the task, the members, and other factors. This begs the question: when should a team favor one team structure over another? Should the team have centralized or decentralized hierarchy? Should it enforce equal participation from each member? Should members offer each other more encouraging or critical feedback? The wrong decisions can doom a team to dysfunction [32, 53, 3, 4]. Even highly-paid experts—managers—struggle to pick effective team structures [15]. They are hardly to blame, as the set of possibilities is vast [29], with lengthy volumes, dedicated

Not all data is good

We found that when multiple people try to teach a robot how to do the same task, the robot tends to be worse at learning the task.

We had a hunch that inconsistent actions in similar situations were the cause of this.

What is the quickest way to test this?

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

Abstract: Imitation learning from human-provided demonstrations is a strong approach for learning policies for robot manipulation. While the ideal dataset for imitation learning is homogenous and low-variance – reflecting a single, optimal method for performing a task – natural human behavior has a great deal of *heterogeneity*, with several optimal ways to demonstrate a task. This multimodality is inconsequential to human users, with task variations manifesting as subconscious choices; for example, reaching *down, then across* to grasp an object, versus reaching *across, then down*. Yet, this mismatch presents a problem for interactive imitation learning, where sequences of users improve on a policy by iteratively collecting new, possibly conflicting demonstrations. To combat this problem of demonstrator incompatibility, this work designs an approach for 1) *measuring the compatibility* of a new demonstration given a base policy, and 2) *actively eliciting more compatible demonstrations* from new users. Across two simulation tasks requiring long-horizon, dexterous manipulation and a real-world "food plating" task with a Franka Emika Panda arm, we show that we can both identify incompatible demonstrations via post-hoc filtering, and apply our compatibility measure to actively elicit compatible demonstrations from new users, leading to improved task success rates across simulated and real environments.[1]

Keywords: Interactive Imitation Learning, Active Demonstration Elicitation, Human-Robot Interaction

1 Introduction

Interactive imitation learning [1, 2, 3] from a pool of human demonstrators is a scalable approach for learning multi-task policies for robotic manipulation [4, 5, 6]. Yet, such approaches have a critical problem, especially in the low-to-moderate data regime: data from multiple human demonstrators often have *conflicting* modes, where two users provide opposing behaviors for a single task – behaviors that manifest as subconscious, random choices. For example, consider the nut-on-peg task in Fig. 1: one user (in orange) approaches the nut by moving *across the table, then down*, while the other user (blue) reaches *down, then across*.

Training on aggregated batches of data in series – starting with a base policy, adding small amounts of data from new users, and retraining the policy after each batch – is common in interactive imitation formulations [2, 3]; unfortunately, when we add a small number of conflicting demonstrations during the interaction phase, the retrained policy attempts to cover both the base demonstrations *and* the new set. This leads to incongruent overfitting, where a policy – even one equipped to learn multimodal behaviors [7, 8, 9] – tries to fit the base set for most of a trajectory, but overfits to the new set for a small subset of the state space, often with catastrophic failure modes.

To mitigate this problem, this work tackles two questions: 1) *how can we measure the compatibility between a new demonstrator and an existing policy*, and 2) *how can we use this measure to actively elicit better demonstrations from a new user?*

While our approach for measuring and eliciting compatible demonstrations *during online collection* is novel, prior work has studied the impact of suboptimal demonstrations on learning. Most relevant, Mandlekar et al. [10] introduce RoboMimic, a suite of simulated manipulation tasks that consist

[1] Additional videos & results: https://sites.google.com/view/eliciting-demos-corl22/home

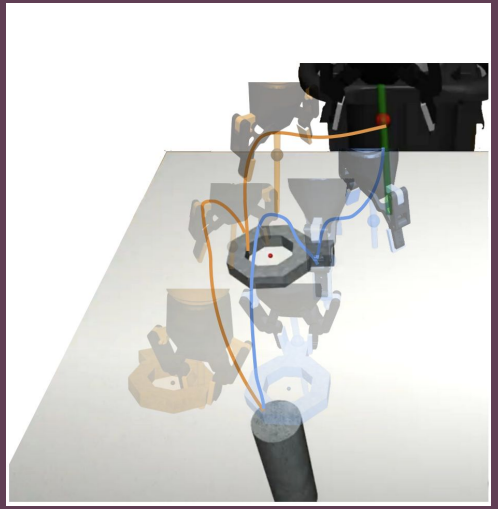
6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

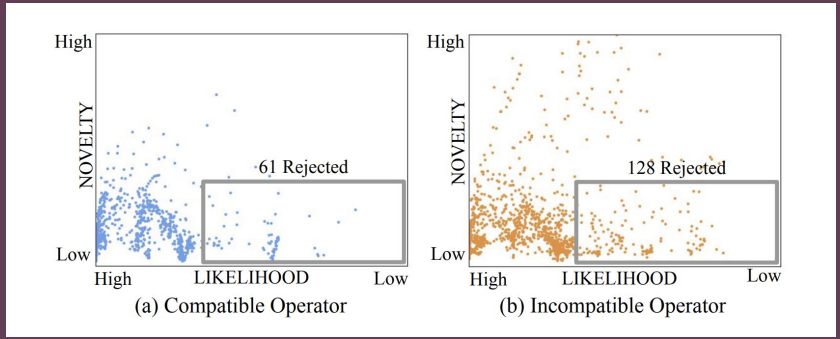
Teaching users to be better teachers

In a 2d maze, the demonstration either went right and then up (RU) or up and then right (UR).

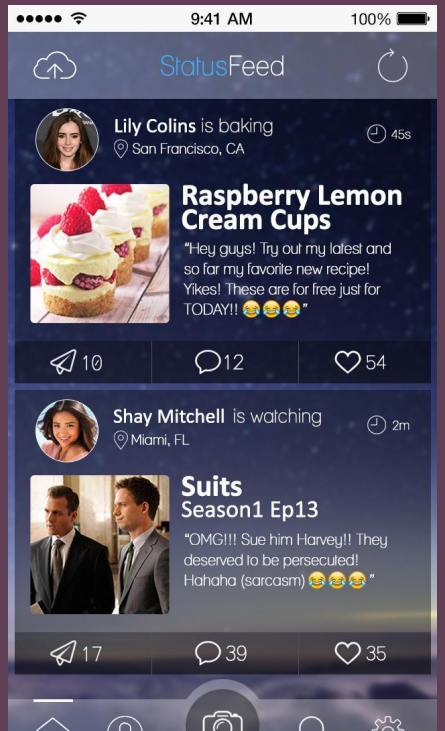
Then I either used all the data, or just one ‘style’ of data.

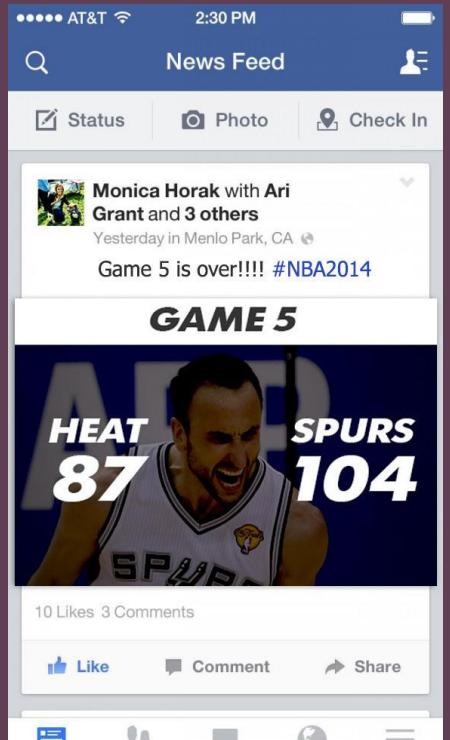
Next: How do we identify ‘bad’ data?

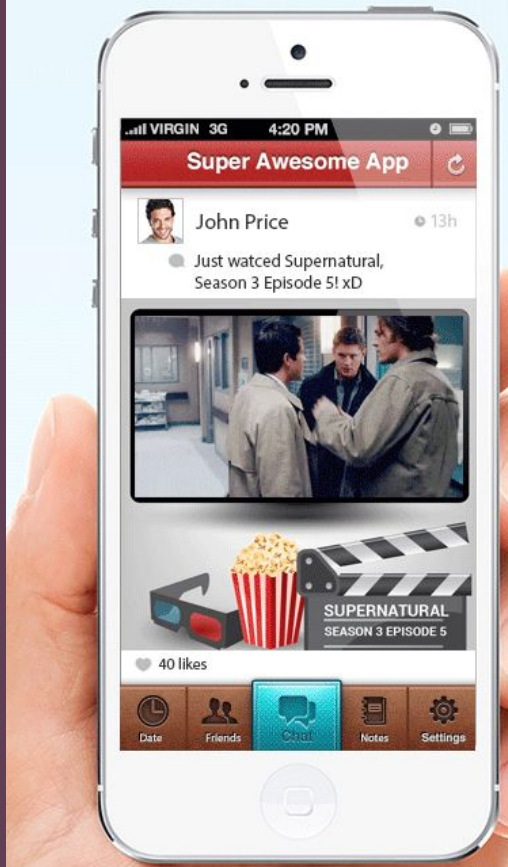




We sketched out a few ideas and then hired Upwork designers to create some mocks of what they might look like. (We decided it wasn't cool enough and dropped the project for the time being.)







theory — piecework

We wanted to understand how the history of p
explain unanswered questions in crowd work:

- Complexity Limits of On–Demand Work
- Decomposing Work
- Workers' Relationships to their Work

And maybe there might be others, we thought?

Does the piecework history help us explain th

Examining Crowd Work and Gig Work Through The Historical Lens of Piecework

Ali Alkhatib, Michael S. Bernstein, Margaret Levi
Computer Science Department and CASBS
Stanford University
{ali.alkhatib, msb}@cs.stanford.edu, mlevi@stanford.edu

ABSTRACT
The internet is empowering the rise of crowd work, gig work, and other forms of on–demand labor. A large and growing body of scholarship has attempted to predict the socio–technical outcomes of this shift, especially addressing three questions: 1) What are the complexity limits of on–demand work?; 2) How far can work be decomposed into smaller microtasks?, and 3) What will work and the place of work look like for workers? In this paper, we look to the historical scholarship on piecework — a similar trend of work decomposition, distribution, and payment that was popular at the turn of the 20th century — to understand how these questions might play out with modern on–demand work. We identify the mechanisms that enabled and limited piecework historically, and identify whether on–demand work faces the same pitfalls or might differentiate itself. This approach introduces theoretical grounding that can help address some of the most persistent questions in crowd work, and suggests design interventions that learn from history rather than repeat it.

ACM Classification Keywords
H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces

Author Keywords
Crowd work; gig work; on–demand work; piecework

INTRODUCTION
The past decade has seen a flourishing of computationally–mediated labor. A framing of work into modular, pre–defined components enables computational hiring and management of workers at scale [68, 17, 83]. In this regime, distributed workers engage in work whenever their schedules allow, often with little to no awareness of the broader context of the work, and often with fleeting identities and associations [104, 94].

For years, such labor was limited to information work such as data annotation and surveys [82, 161, 166, 51, 119]. However, physically embodied work such as driving and cleaning have now spawned multiple online labor markets as well [94, 3, 1, 2]. In this paper we will use the term *on–demand labor*, to capture this pair of related phenomena: first, *crowd work* [83], on platforms such as Amazon Mechanical Turk (AMT) and other sites of (predominantly) information work; and second, *gig work* [48, 118], often as platforms for one–off jobs, like driving, courier services, and administrative support.

The realization that complex goals can be accomplished by directing crowds of workers has spurred firms to explore sites of labor such as AMT to find the limits of this distributed, on–demand workforce. Researchers have also taken to the space in earnest, developing systems that enable new forms of production (e.g. [14, 18, 117]) and pursuing social scientific inquiry into the workers on these platforms [128, 138]. This research has identified the sociality of gig work [54], as well as the frustration and disenfranchisement that these systems effect [72, 104, 106]. Others have focused on the responses to this frustration, reflecting on the resistance that workers express against digitally–mediated labor markets [94, 133].

This body of research has broadly worked toward the answer to one central question: *What does the future hold for on–demand work and those who do it?* Researchers have offered insights on this question along three major threads: First, what are the complexity limits of on–demand work — specifically, how complex are the goals that crowd work can accomplish, and what kinds of industries may eventually utilize it [142, 79, 165, 164, 110, 59]? Second, how far can work be decomposed into smaller microtasks [27, 100, 92, 29, 111]? And third, what will work and the place of work look like for workers [72, 73, 54, 106]?

This research has largely sought to answer these questions by examining extant on–demand work phenomena. So far, it has not offered an ontology to describe or understand the developments in worker processes that researchers have developed, or the emergent phenomena in social environments; nor has any research gone so far as to anticipate future developments.

Piecework as a lens to understand on–demand work
In this paper, we offer a framing for on–demand work as a contemporary instantiation of *piecework*, a work and payment structure which breaks tasks down into discrete jobs, wherein payment is made for *output*, rather than for *time*. We are not the first to relate on–demand work to piecework; in 2013, for

CHI 2017, May 06 – 11, 2017, Denver, CO, USA

ISBN 978-1-4503-4655-9/17/05…$15.00
DOI http://dx.doi.org/10.1145/3025453.3025974

theory — piecework

Do a quick exploration of each question. Try writing a short white paper for it — less than a page. Aim to write three or more.

Don't worry about final quality. Our goal is to mainly see if "there's a there there": if it's interesting enough to go deeper.

Examining Crowd Work and Gig Work Through The Historical Lens of Piecework

Ali Alkhatib, Michael S. Bernstein, Margaret Levi
Computer Science Department and CASBS
Stanford University
{ali.alkhatib, msb}@cs.stanford.edu, mlevi@stanford.edu

ABSTRACT

The internet is empowering the rise of crowd work, gig work, and other forms of on–demand labor. A large and growing body of scholarship has attempted to predict the socio–technical outcomes of this shift, especially addressing three questions: 1) What are the complexity limits of on–demand work?, 2) How far can work be decomposed into smaller microtasks?, and 3) What will work and the place of work look like for workers? In this paper, we look to the historical scholarship on piecework — a similar trend of work decomposition, distribution, and payment that was popular at the turn of the 20th century — to understand how these questions might play out with modern on–demand work. We identify the mechanisms that enabled and limited piecework historically, and identify whether on–demand work faces the same pitfalls or might differentiate itself. This approach introduces theoretical grounding that can help address some of the most persistent questions in crowd work, and suggests design interventions that learn from history rather than repeat it.

ACM Classification Keywords

H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces

Author Keywords

Crowd work; gig work; on–demand work; piecework

INTRODUCTION

The past decade has seen a flourishing of computationally–mediated labor. A framing of work into modular, pre–defined components enables computational hiring and management of workers at scale [68, 17, 83]. In this regime, distributed workers engage in work whenever their schedules allow, often with little to no awareness of the broader context of the work, and often with fleeting identities and associations [104, 94].

For years, such labor was limited to information work such as data annotation and surveys [82, 161, 166, 51, 119]. However, physically embodied work such as driving and cleaning have now spawned multiple online labor markets as well [94, 3, 1, 2]. In this paper we will use the term *on–demand labor*, to capture this pair of related phenomena: first, *crowd work* [83], on platforms such as Amazon Mechanical Turk (AMT) and other sites of (predominantly) information work; and second, *gig work* [48, 118], often as platforms for one–off jobs, like driving, courier services, and administrative support.

The realization that complex goals can be accomplished by directing crowds of workers has spurred firms to explore sites of labor such as AMT to find the limits of this distributed, on–demand workforce. Researchers have also taken to the space in earnest, developing systems that enable new forms of production (e.g. [14, 18, 117]) and pursuing social scientific inquiry into the workers on these platforms [128, 138]. This research has identified the sociality of gig work [54], as well as the frustration and disenfranchisement that these systems effect [72, 104, 106]. Others have focused on the responses to this frustration, reflecting on the resistance that workers express against digitally–mediated labor markets [94, 133].

This body of research has broadly worked toward the answer to one central question: *What does the future hold for on–demand work and those who do it?* Researchers have offered insights on this question along three major threads: First, what are the complexity limits of on–demand work — specifically, how complex are the goals that crowd work can accomplish, and what kinds of industries may eventually utilize it [142, 79, 165, 164, 110, 59]? Second, how far can work be decomposed into smaller microtasks [27, 100, 92, 29, 111]? And third, what will work and the place of work look like for workers [72, 73, 54, 106]?

This research has largely sought to answer these questions by examining extant on–demand work phenomena. So far, it has not offered an ontology to describe or understand the developments in worker processes that researchers have developed, or the emergent phenomena in social environments; nor has any research gone so far as to anticipate future developments.

Piecework as a lens to understand on–demand work

In this paper, we offer a framing for on–demand work as a contemporary instantiation of *piecework*, a work and payment structure which breaks tasks down into discrete jobs, wherein payment is made for *output*, rather than for *time*. We are not the first to relate on–demand work to piecework: in 2013, for

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ISBN 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025974

Main Take away

Once a direction of highest risk is chosen (Vector)

What is the quickest way to learn about the idea?

Prototype the periphery, choose the easiest task

Focus on the core

Let’s Try It

Your turn

Pair up with someone not on your project.

5 min each person: describe your project's current state, the current question you're trying answer. Brainstorm together how to increase velocity.

Afterwards, we'll share out.

A reminder: the algorithm

1. Articulate the question you're answering (vector).
2. Decide what's absolutely core to answering that question.
3. Decide what's peripheral.
4. Decide the level of fidelity that is absolutely necessary.
5. Go — but be open to reevaluating your assumptions as you go.
6. Loop with a new question.

Tips and tricks

“I’m being low velocity.”

Velocity = distance / time

So, if your velocity is low, you have two options:

1. Cover more distance: habits that can get you further in the same time (e.g., “try harder”, “be a better engineer”)

You’re typically already maxed out on this.

2. Decrease the time: prototype more effectively

WIN. Prototype more narrowly, lower your fidelity expectations (e.g., spit out any draft)

“I’m being low velocity.”

Velocity = distance / time, if your velocity/learning is low, you can:

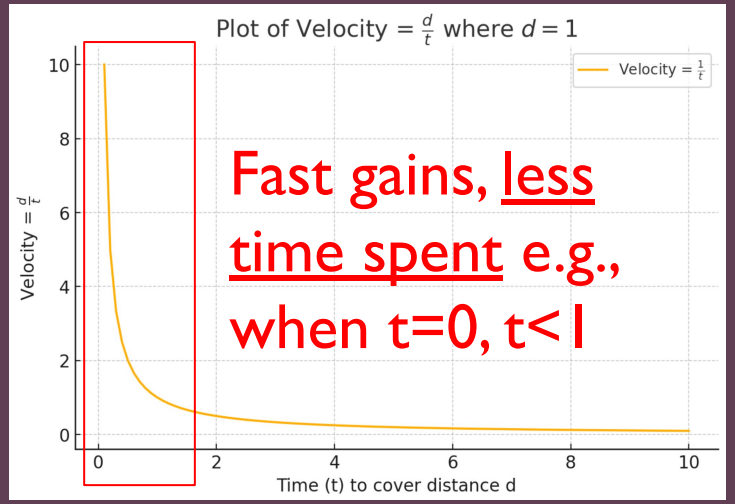
1. Cover more distance: Only ~linear gains given fix time spent
2. Decrease the time: gives fast gains, especially early on!
   ’(eventually it does plateau)

fastest is to not do [t=0] → ∞

or do quickly [t<1] (steep!)

Extreme1: you get infinite velocity, t=0 suggests don’t do periphery if you can!

Extreme2: less time t<1 → faster



On Tiktok or Twitter or E-mail… ?

This signals a lack of focus, and is a pretty certain predictor that you're in a swamp.

It means you're prototyping too broadly: you're unfocused! focus your goal.

Or you're requiring too high a level of fidelity: you have unreasonable standards! lower your expectations.

Develop an internal velocity sensor, and as soon as you recognize this, apply one of the two rules.

Focus or lower fidelity

Lowering standards: parallelism

Too often, we suffer from what's known in the literature as fixation: being certain in an idea and pursuing it to the exclusion of all else. We cannot separate ego from artifact.

Instead, to answer the question, it's often best to explore multiple approaches in parallel.

"While the quantity group was busily churning out piles of work—and learning from their mistakes—the quality group had sat theorizing about perfection, and in the end had little more to show for their efforts than grandiose theories and a pile of dead clay."
— Bayles and Orland, 2001

Corollary 1: pivoting

Velocity is why cutting yourself off short and pivoting to a new project can be so dangerous in research.

Typically people pivot after a week in the swamp (the "fatal flaw fallacy"), rather than iterating with high velocity out of the swamp.

I promise that the project you pivot to will have a swamp too.

Learn to increase velocity and prototype your way out of the swamp faster, instead of seeking out a swampless project.

Corollary 2: technical debt

Technical debt := "cost of taking too many shortcuts"
Obviously, at some point you need to make sure you're not too deep in technical debt, design debt, or writing debt.

But luckily, most people can only run their processors hot for a few hours a day. Everything I've described takes a lot out of you.

When you're out of creative cycles, spend time maturing other parts of your project that are no longer open questions (help time [t] decrease later). Or, sometimes we reach a phase where we pause prototyping and focus on refinement and execution for a bit.

Tip: Talking to others/presenting in lab can help in creativity too!

Corollary 3: More tips

Tip: walks with no headphones

1. You can be more creative on a fast prototype (velocity)
2. You can be more creative to think of possible unknowns (vector)
3. You can even refine your attempt when you are commuting

Reflect often on what you learned and how you could have been more aggressive to prototype

Honesty: Was that really necessary? What did I truly get from this?

Mindset: Failure is good! Because learning is good!

Why is velocity so important?

Great research requires high velocity

Don’t let 6-12 month paper deadlines obscure the velocity at which research needs to move in order to succeed.

If you want to achieve a high impact idea, you need to try a lot of approaches and refine and fail a lot. You want to do that as quickly as possible due to uncertainty.

If you can prototype and learn and fail 5x as quickly as the next person, you will be able to achieve far more (de) risky and impactful research.

Takeaways, in brief

1) The swamp is real, and it slows visible progress.

2) Velocity is a far better measure/metric of yourself than progress, and it's something you actually have control over.
(you can't control experiments working in unknown envs)

3) Achieve high velocity by being clear what question you're answering, and focusing ruthlessly on the core of that question while stripping out the periphery.

4) If you're low velocity, velocity = distance / time. Either increase distance (rarely possible) or decrease time (often possible: you're too broad or too perfectionist or doing too much.

And finally…

Get into your project groups and discuss your strategy for velocity. What's working? What can be improved?

Velocity in Research