Evaluation

(I forgot to include the researcher of the week video since I was at CHI, it will be back next week!)
Congrats on finishing A3!

Similar to A2, get 50% points back if revisions by Mon 5/16

Evaluation plan assignment going live today, due in one week

Includes submitting 2 questions for the panelists

Research careers panel next week!
Research careers panel

Cody Coleman

Michelle Lee

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

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

**Velocity**: rapid reduction of risk in the chosen dimension
“But how would we even evaluate that?”

People often rush to this question early on in ideation.

Today’s goal is to provide scaffolding for how to answer it.
Today’s big idea: evaluation

How do we get precise about what we need to evaluate for our project?
How do we design an appropriate evaluation?
How do we analyze our evaluation results?
Why perform evaluation in research?
Recall from Week 1 that research introduces a new idea into the world. So...how do we know if that idea is worth adopting or paying attention to?

Option 1 ("The Simon Cowell Solution"): Academia’s Got Talent, Shark Tank — Convince people your idea is The Best One

Option 2: Construct an evaluation to test the idea fairly

Let's do this one: the goal isn't advocacy; it's an understanding of the idea's strengths and limits
Standards of evidence

Every field has an accepted standard of evidence — a set of methods that are agreed upon for proving a point

- Medicine: Double-blind randomized controlled trial
- Philosophy: Rhetoric
- Math: Formal proof
- Applied Physics: Measurement
Standards of evidence

In computing, because areas use different methods, the standard of evidence differs based on the area.

Your goal: convince an expert in your area.

So, use the methods appropriate to your area.
Designing an evaluation
In-class activity

Today, you will outline how to evaluate your idea and bit flip.

The expectation is **not** that you are already ready to evaluate your system!

Some fields naturally have constant, continual evaluation baked into the research process. Others have front-loaded design or engineering costs.

However, knowing how you will evaluate your project will help you better determine what is core and what is periphery in what you build.

So...think big and high-level for this activity!

You will submit this outline as part of Assignment 4.

http://tiny.cc/cs197-eval
Problematic point of view

“But how would we evaluate this?”

Why is this point of view problematic?

Implication: “I believe the idea is right, but I don’t believe that we can prove it.”

Implication: “The thread of designing the evaluation is different than the process. Evaluation is distinct from the validity of the idea.”

Neither implication is correct. **If you can precisely articulate your idea and your bit flip, then you can design an appropriate evaluation.**

**If you can’t precisely articulate your idea and your bit flip, then you can’t design an appropriate evaluation.**
Step 1: articulate your thesis

A much more productive approach is to derive an evaluation design directly from your idea.

What is the main thesis of your work?

(Lucky for you, you came up with this when writing the Introduction of your paper. It’s the topic sentence of your bit flip paragraph.)
Recall:

**Bit**

Network behaviors are defined in hardware, statically.

Code compilers should utilize smart algorithms to optimize into machine code.

A minimum graph cut algorithms should always return correct answers.

**Flip**

If we define the behaviors in software, networks can become dynamic and more easily debuggable.

Code compilers will find more efficient outcomes if they just do monte carlo (random!) explorations of optimizations.

A randomized, probabilistic algorithm will be much faster, and we can still prove a limited probability of an error.
Recall:

Bit
Network behaviors are defined in hardware, statically.

Code compilers should utilize smart algorithms to optimize into machine code.

A minimum graph cut algorithms should always return correct answers.

Flip
If we define the behaviors in software, networks can become dynamic and more easily debuggable.

Code compilers will find more efficient outcomes if they just do monte carlo (random!) explorations of optimizations.

A randomized, probabilistic algorithm will be much faster, and we can still prove a limited probability of an error.

[3 min] Step 1. On your slide, write your thesis as A) a bit and B) flip.

http://tiny.cc/cs197-eval
Step 2: map your thesis onto a claim

There are only a small number of claim structures implicit in most theses:

- $x > y$: approach $x$ is better than approach $y$ at solving the problem
- $\exists x$: it is possible to construct an $x$ that satisfies some criteria, whereas it was not possible before
- **bounding** $x$: approach $x$ only works given certain assumptions
<table>
<thead>
<tr>
<th><strong>Bit</strong></th>
<th><strong>Flip</strong></th>
<th><strong>Claim</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Network behaviors are defined in hardware, statically.</td>
<td>If we define the behaviors in software, networks can become dynamic.</td>
<td>∃ x: software-defined behaviors can be changed on the fly</td>
</tr>
<tr>
<td>Code compilers should utilize smart algorithms to optimize into machine code.</td>
<td>Code compilers will find more efficient outcomes if they just do monte carlo (random!) explorations of optimizations.</td>
<td>x &gt; y: monte carlo exploration will produce more optimized code than hand-tuned compilers</td>
</tr>
<tr>
<td>A minimum graph cut algorithms should always return correct answers.</td>
<td>A randomized, probabilistic algorithm will be much faster, and we can still prove a limited probability of an error.</td>
<td>x &gt; y: a randomized graph cut algorithm is faster and has bounded error</td>
</tr>
<tr>
<td>Bit</td>
<td>Flip</td>
<td>Claim</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Network behaviors are defined in hardware,</td>
<td>If we define the behaviors in software, networks can become dynamic.</td>
<td>(\exists x: \text{software-defined behaviors can be changed on the fly} )</td>
</tr>
<tr>
<td>statically.</td>
<td></td>
<td>(x &gt; y: \text{monte carlo exploration will produce more optimized code than hand-tuned compilers})</td>
</tr>
<tr>
<td>Code compilers should utilize smart algorithms to optimize into machine code.</td>
<td>Code compilers will find more efficient outcomes if they just do monte carlo (random!) explorations of optimizations.</td>
<td></td>
</tr>
<tr>
<td>A minimum graph cut algorithms should</td>
<td>A randomized, probabilistic algorithm will be much faster,</td>
<td>(x &gt; y: \text{a randomized graph cut algorithm is faster and has bounded error})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[3 min] Step 2. On your slide, identify your claim type, and C) write your claim.
Step 3: claims imply an evaluation design

Each claim structure implies an evaluation design:

- \( x > y \): given a representative task or set of tasks, test whether \( x \) in fact outperforms \( y \) at the problem
- \( \exists x \): demonstrate that your approach achieves \( x \)
- **bounding** \( x \): demonstrate bounds inside or outside of which approach \( x \) fails
<table>
<thead>
<tr>
<th>Flip</th>
<th>Claim</th>
<th>Implied evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>If we define the behaviors in software, networks can become dynamic.</td>
<td>( \exists x: \text{software-defined behaviors can be changed on the fly} )</td>
<td>Demonstrate that behaviors propagate, and which kind of behaviors can be authored</td>
</tr>
<tr>
<td>Code compilers will find more efficient outcomes if they just do monte carlo (random!) explorations of optimizations.</td>
<td>( x &gt; y: \text{monte carlo exploration will produce more optimized code than hand-tuned compilers} )</td>
<td>Compare runtime of generated machine code against known best approaches</td>
</tr>
<tr>
<td>A randomized, probabilistic algorithm will be much faster, and we can still prove a limited probability of an error.</td>
<td>( x &gt; y: \text{a randomized graph cut algorithm is faster and has bounded error} )</td>
<td>Prove runtime for randomized algorithm (vs. prior algorithm) and probability of error</td>
</tr>
<tr>
<td>Flip</td>
<td>Claim</td>
<td>Implied evaluation</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>If we define the behaviors in software, networks can become dynamic.</td>
<td>$\exists x$: software-defined behaviors can be changed on the fly</td>
<td>Demonstrate that behaviors propagate, and which kind of behaviors can be authored</td>
</tr>
<tr>
<td>Code compilers will find more efficient outcomes if they just do monte carlo (random!) explorations of optimizations.</td>
<td>$x &gt; y$: monte carlo exploration will produce more optimized code than hand-tuned compilers</td>
<td>Compare runtime of generated machine code against known best approaches</td>
</tr>
<tr>
<td>A randomized, probabilistic algorithm will be much faster, and we will see limited deviation.</td>
<td>$x &gt; y$: a randomized graph cut algorithm is faster and has bounded error</td>
<td>Prove runtime for randomized algorithm (vs. prior algorithm) and probability of error</td>
</tr>
</tbody>
</table>

[3 min] Step 3. On your slide, write the high-level evaluation design.
Share out: breakout rooms
(6 min, 2 min each)
Architecture of an Evaluation
Four constructs that matter

To develop your evaluation plan, you need to get precise about four components of your evaluation:

- Dependent variable
- Independent variable
- Task
- Threats
In other words, what's the **outcome** you're measuring?


The choice of this quantity should be clearly implied by your thesis.

It’s often tempting to measure many DVs, and I'm not against doing so. However, one should be your central outcome, and the others auxiliary.
**IV: independent variable**

In other words, what determines what $x$ and $y$ are? What are you manipulating in order to cause the change in the dependent variable?

The IV is the construct that leads to conditions in your evaluation. Examples might include:

- Algorithm
- Dataset size or quality
- Interface
### Example: Clothing rigs

<table>
<thead>
<tr>
<th>Bit</th>
<th>Flip</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D drawings of clothes for mix &amp; match characters (e.g., Bitmoji) are independent from the body, so they become separated if the body changes.</td>
<td>We can create rigs that automatically connect the clothing and body layers so they move together.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Claim</th>
<th>Implied evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \exists x ) Rigs to attach clothes the body</td>
<td>Demonstrate that rigs do indeed work by showing lots of examples.</td>
</tr>
</tbody>
</table>
Example: Clothing rigs

**Bit**  2D drawings of clothes for mix & match characters (e.g., Bitmoji) are independent from the body, so they become separated if the body changes

**Claim** $\exists \mathbf{x}$ Rigs to attach clothes the body

**Flip** We can create rigs that automatically connect the clothing and body layers so they move together

**Implied evaluation** Demonstrate that rigs do indeed work by showing lots of examples

**DV** Quality and scope of results

**IV** Kinds of artwork chosen
Example: Clothing rigs

Bit
2D drawings of clothes for mix & match characters (e.g., Bitmoji) are independent from the body, so they become separated if the body changes.

Claim
∃ x Rigs to attach clothes the body

Implied evaluation
Demonstrate that rigs do indeed work by showing lots of examples.

DV
Quality and scope of results

IV
Kinds of artwork chosen

[3 min] In the next slide, write your D) dependent variable and E) independent variable.
Task

What, specifically, is the routine being followed in order to manipulate the independent variable and measure the dependent variable?

We will perform one-shot prediction of classes at the 25th percentile of popularity in ImageNet according to Google search volume.

Participants will have thirty seconds to identify each article as disinformation or not, within-subjects, randomizing across interfaces.

We will run a performance benchmark drawn from Author et al. against each system.
Example: Clothing rigs

Bit 2D drawings of clothes for mix & match characters (e.g., Bitmoji) are independent from the body, so they become separated if the body changes.

Claim \( \exists x \) Rigs to attach clothes to the body.

DV Quality and scope of results

Task We will pick a wide variety of artwork from existing mix & match systems to test our rigs on. They will vary across 1) pose and proportion of body and 2) kinds of clothing.

Flip automatically connect the clothing and body layers so they move together.

Implied evaluation Demonstrate that rigs do indeed work by showing lots of examples.

IV Kinds of artwork chosen
Threats

What are your threats to validity? In other words, what might bias your results or mean that you’re telling an incomplete story?

Might your selection of which classes to predict influence the outcome?

Are you running on particular cloud architectures that are amenable to, or not amenable to, your task?

Are your participants biased toward healthy young technophiles?

Do your participants always see the best interface first?
Threats

There are typically three ways to handle these kinds of issues:

1) Argue as irrelevant: yes, that bias might exist, but it’s not conceptually important to the phenomenon you’re studying and is unlikely to strongly affect the outcome or make the results less generalizable

2) Stratify: re-run your evaluation in each setting to see whether the outcomes change

3) Randomize: explicitly randomize (e.g., people) across values of the control variable. For example, randomize the order in which people see the interface.
Example: Clothing rigs

**DV** Quality and scope of results

**Task** We will pick a wide variety of artwork from existing mix & match systems to test our rigs on. They will vary across 1) pose and proportion of body and 2) kinds of clothing.

**Threat** Hand picked examples too much

How do you know capabilities of the system match user desires?

- **IV**

[6 min] In breakout rooms, discuss threats and how to address them. Write in slide.

- **Stratify:** Show on very large dataset

Run a user study to observe the kinds of modifications they make
Find your inspiration

There’s no need to start from scratch on this.

Your nearest neighbor paper, and the rest of your literature search, has likely already introduced evaluation methods into this literature that can be adapted to your purpose.

**Start at your NN paper:** figure out what the norms are, and tweak them. Talk to your mentor, and there are some example papers on the assignment spec.
Your To-dos

1. Continue to update logs

2. Proposed solution & evaluation plan, due 1 week

**Proposed Solution:**
- Describe the high-level and low-level ideas of your system
- Examples on website, but write the minimum needed for a reader to understand the thesis in your Evaluation Plan
- Written (400-600 words)

**Evaluation plan:**
- Thesis, claim, eval design (what we did in class today)
- Outline only

Exit ticket: [http://tiny.cc/cs197-week6](http://tiny.cc/cs197-week6)
Computer Science Research

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