

Communicating Research

CS 197 | Stanford University | Michael Bernstein

Special thanks to Kayvon Fatahalian and Lauren Gillespie for slides

Administrivia

Draft talk is **due in section** the Tuesday after Thanksgiving

Six minutes

You'll be getting feedback from each other and from the TA

Draft talk due in week 10, final paper and talk due during finals

Today's goals

How to structure and deliver a stellar research talk

So, you're giving a talk.

Talks are how people get to know your work

A lot of people might scroll past your social media post about your paper

Few are masochistic enough to scroll through every paper in the archives to run into yours

People might not make it to your poster

But, **we have a culture of showing up to and listening to talks.**

Clarity is highly prized

Great talks get work more attention, adoption, and conversation

“Hey, did you see X’s talk yesterday?”

“Hey, great talk—we should chat about an internship.”

It's a huge shame if you spend months on the project, but can't go the last mile to communicate it to others

Michael's take: one **difference between good and great researchers** is being able to explain how and why the work matters

And now, principles and
tips

Tip I

Identify your one
narrative goal

What do they need to know?

Aim for the audience to understand everything you're saying, and to **feel smarter** at the end of the talk

“Aha! I understand something that I didn’t understand before!”

Even if you are targeting experts in your area, they have not been spending months thinking about this topic

Every step of the talk should be leading them up to an understanding, or helping them understand the consequences of that understanding

A talk is not a step-by-step report—it is an **explainer**

What is the one idea they should walk away with?

You have spent a lot of time in the guts of your project, and know all the details and the caveats and the webs of relationships. Unfortunately, the audience will remember none of that

They will **remember one moment—one idea—from your talk.**
What will that be?

“This claim.”

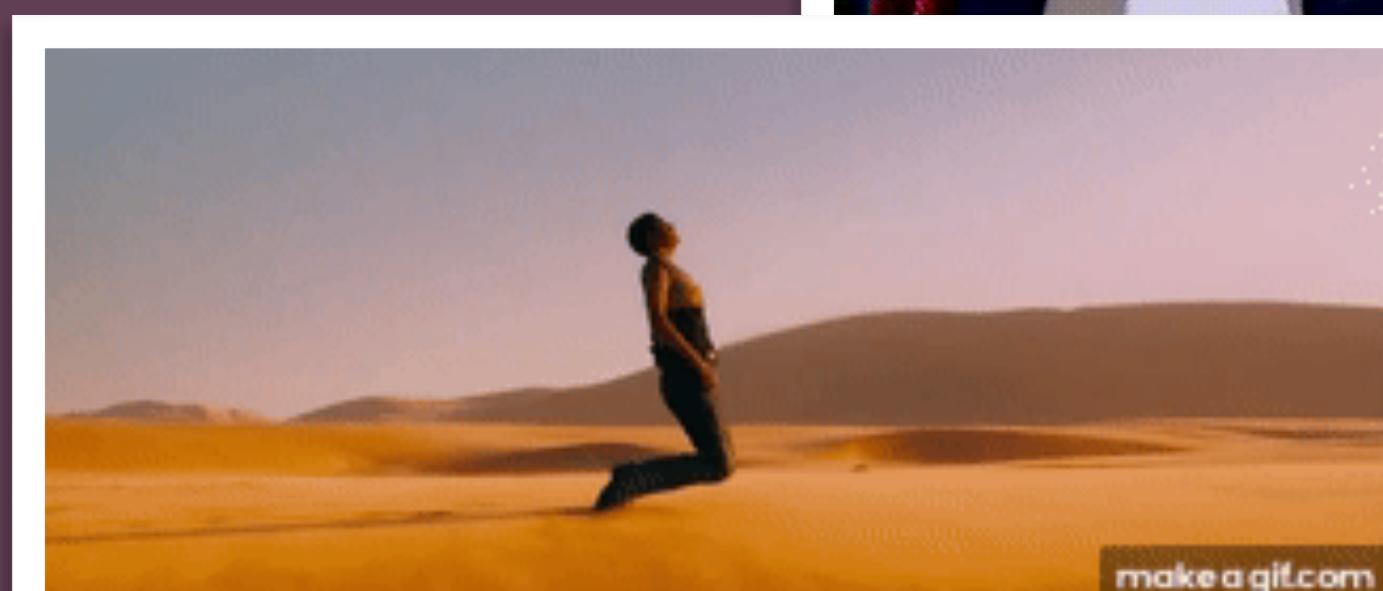
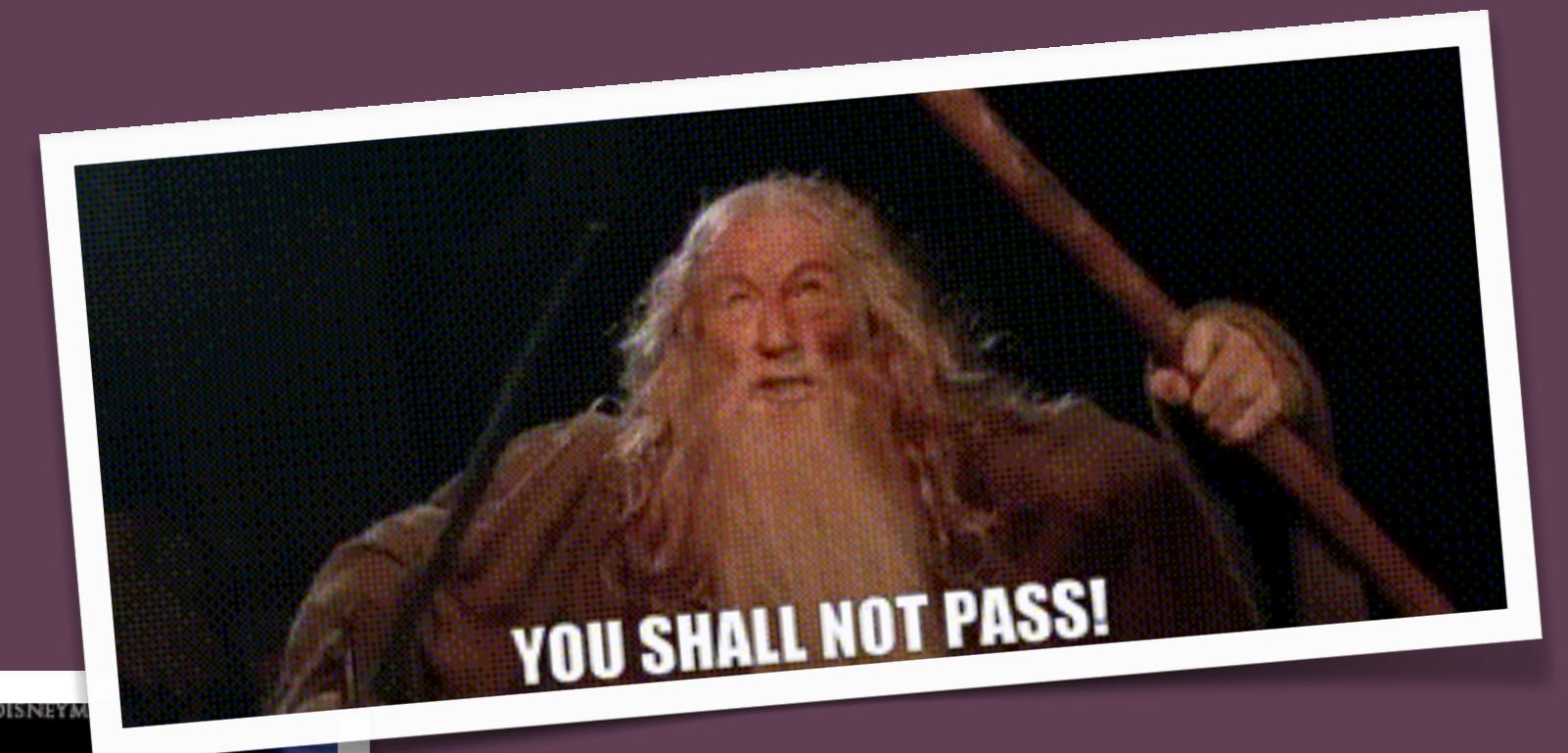
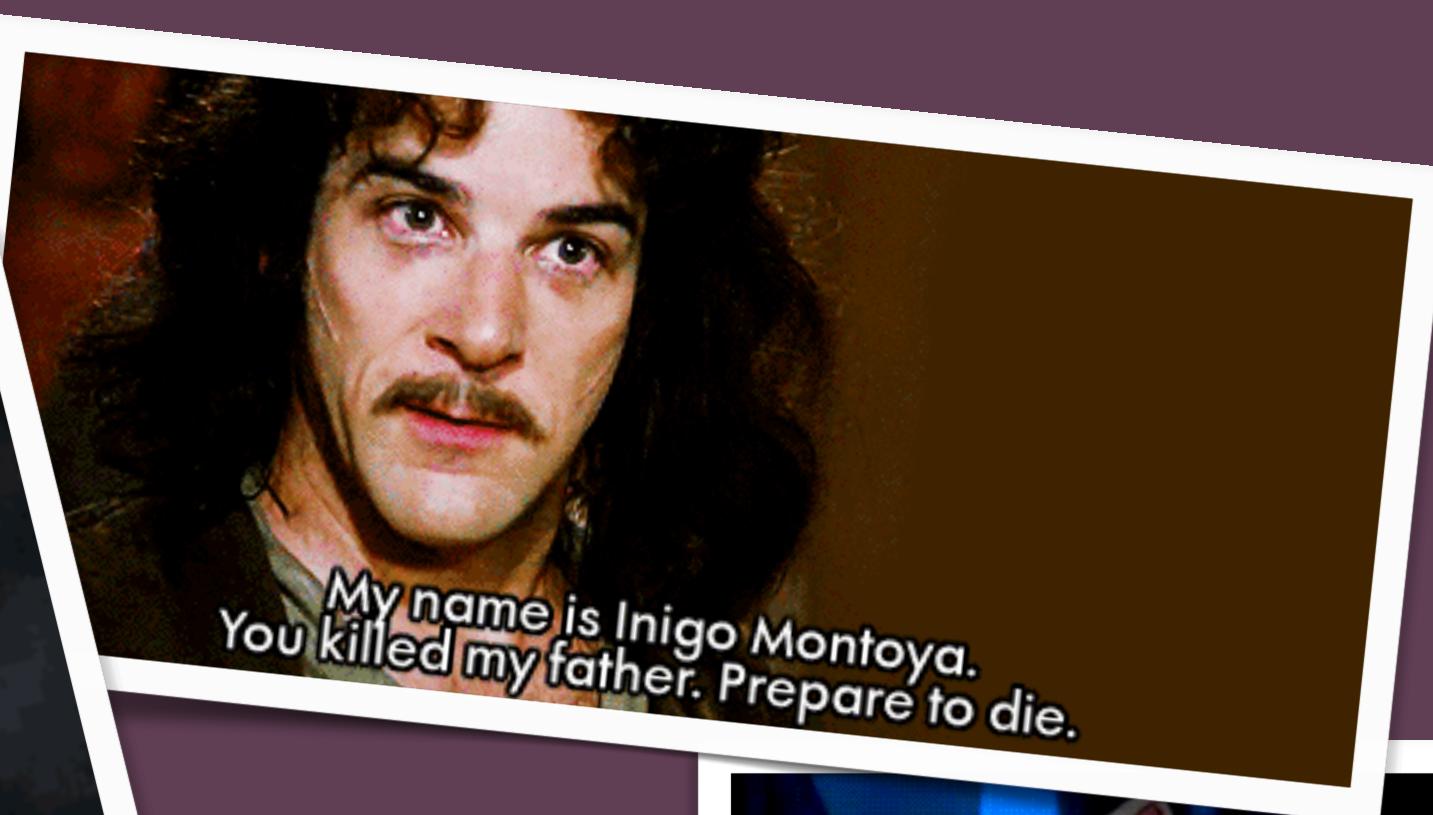
“This diagram.”

“This graph.”

“This result.”

Movies do this too

What is the one image or moment that the whole movie was built around?



Work backwards from your one idea to design the talk

Once you know what the high point of your talk will be, ask yourself:

What is necessary (and what's not) to lead up to it?

How do I highlight that it's the high point so that they remember it?

How do I hit a denouement afterwards?

This means that **your talk will likely not cover everything**

Tip II

Every sentence matters.

Intros are often very poor

“LLMs have been shown to have good task performance on a variety of tasks...” Everyone knows this. So don’t say it.

Unlike a paper, related work in a talk does not exist for academic completeness

The goal of the intro and background is to tell the listener: “**In this talk, here is the way I want you to think about the problem I am trying to solve.**”

Bad example I

Never do this:

All I now
know is that
you will have
the talk outline
I would have
already
expected

The image shows a presentation slide with a blue header bar containing the word "Outline". Below the header is a list of bullet points. The first four items are bolded, while the last two are regular text. The third item under "Proposed System Architecture" has three sub-points indicated by diamond symbols.

- **Introduction**
- **Related Work**
- **Proposed System Architecture**
 - ❖ Basic design decision
 - ❖ Dedicated hardware for T&I
 - ❖ Reconfigurable processor for RGS
- **Results and Analysis**
- **Conclusion**

Bad example 2

What is the idea or bit flip that you're building towards?

 **Discrete Methods**

- Remove/Add Pixels
- Related work*:
 - Seam Carving [SIGGRAPH 07]
 - Improved Seam Carving [SIGGRAPH 08]
 - Shift-Map Image Editing [ICCV 09]

* non-exhaustive



From: Avidan et.al., Seam Carving for Content-Aware Image Resizing



From: Pritch et.al., Shift-Map Image Editing

Experts already know these papers exist, and non-experts aren't going to learn these papers from seeing their names

This type of related work section says little more than “others have worked in this area before”. I suspect your audience already assumes this is the case.

 **Continuous methods**

- Find continuous transformation
- Warp/deformation grid
- Related work*
 - Non-homogenous warping, ICCV 07
 - Streaming video, SIGGRAPH 09
 - Shrinkability Maps for Content-Aware Video Resizing, PG 09
 - Robust Image Retargeting via Axis-Aligned Deformation, EG 12



Generated with the Streaming Video approach [KRA09]



original saliency map



retargeting to 200% with using axis-aligned deformations

From: Robust Image Retargeting via Axis-Aligned Deformation

“Think of it this way:”

The goal of the intro and background is to tell the listener: “During this talk, here is the way I want you to think about the problem I am trying to solve.”

An excellent strategy to catch the audience’s attention and frame the story is to make them aware that there is something they didn’t know they didn’t know.

A better example [Mullapudi 2019]

The talk intro you're about to see asked the audience to consider the following question:

Why are we working so hard to train general ML models that work in many different situations?

Why don't we just use simple, less general ML models and retrain them constantly for the specific conditions at hand?

Here's the sequence used to do that...

What a traffic camera sees



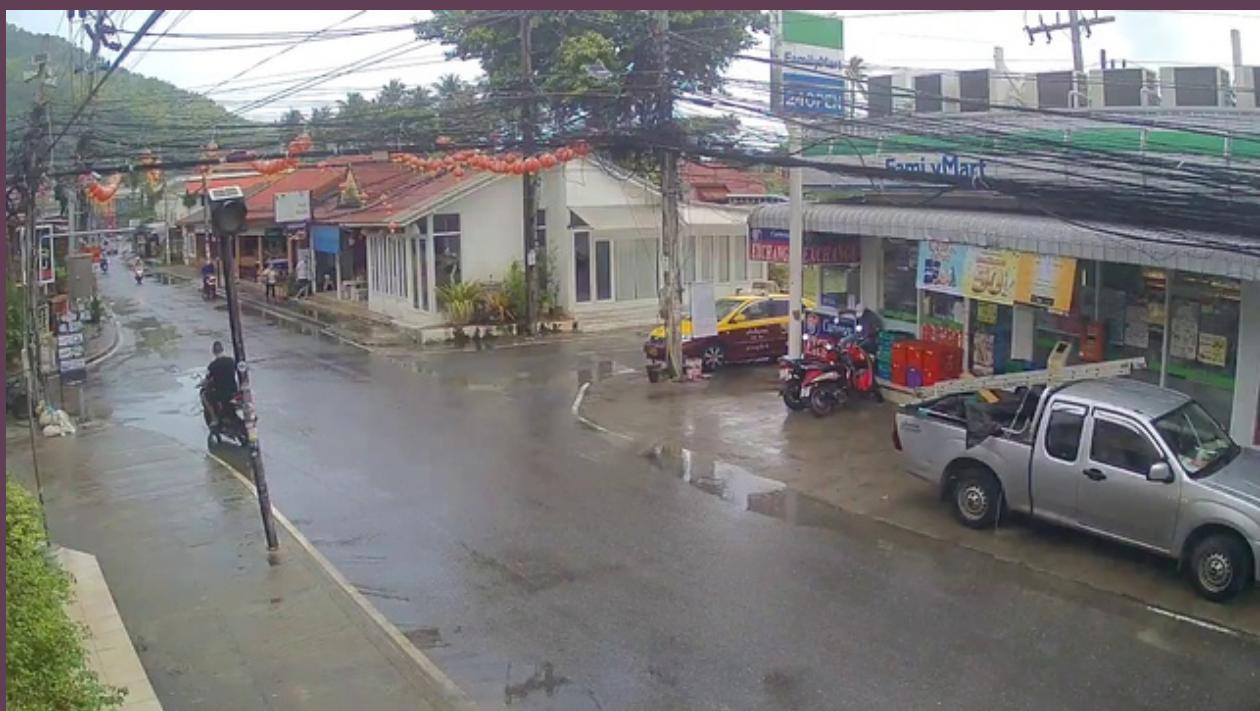
Jackson Hole Town Square @ Pizzeria Caldera 12/13/2017 01:20:28 PM

It seems like it should be easy to train a simple, low-cost ML model to detect cars/ pedestrians/etc in this one scene...right?

Problem: distribution shift



But any one scene can change dramatically due to weather, time-of-day, types of vehicles, etc...



We need to acquire a diverse training set...right?

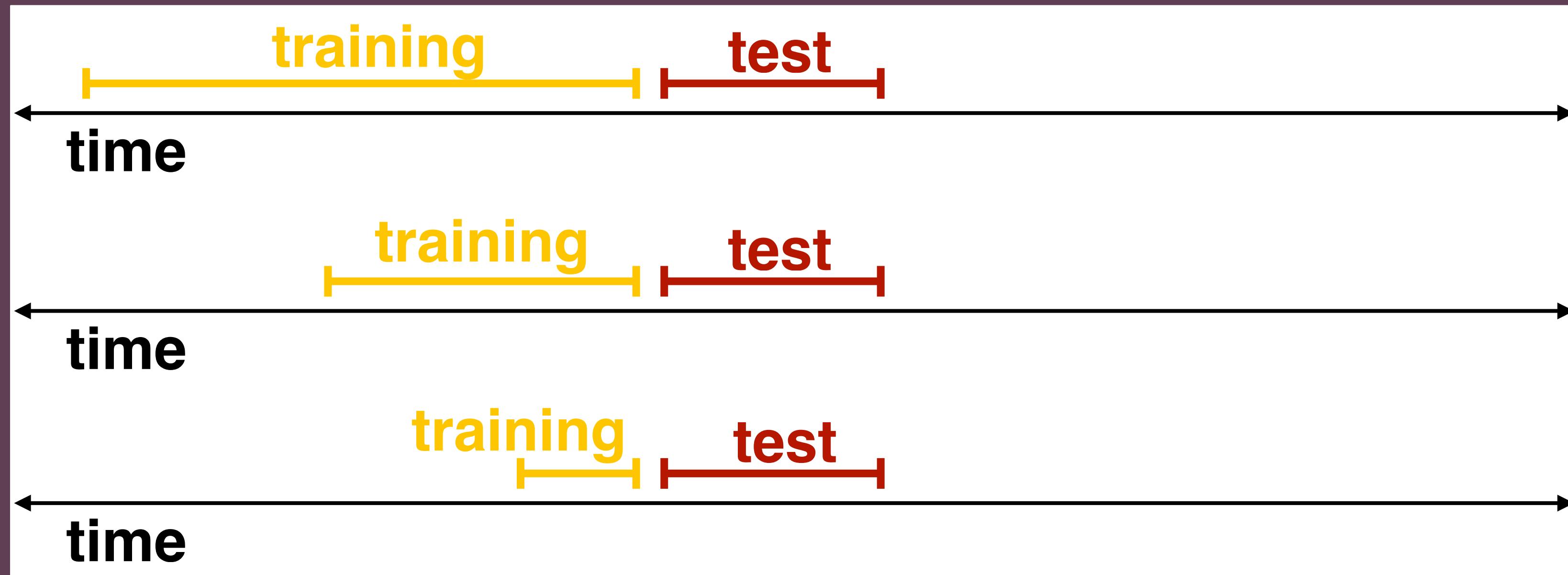


Consider this experiment

Plop camera down in a new environment

We want a specialized model for processing the stream from this camera

How much training data is needed to train an accurate model?



Now the audience is thinking, and ready for you to help them feel smart for figuring it out



Continuous adaptation!

The premise: low cost efficient models can retain high accuracy for complex tasks in challenging environments if they are continuously specialized to the contents of video streams

a.k.a. Don't worry about carefully curating the perfect training set up front, just make sure you can adapt quickly online when you see it

Audience member: "But how do I adapt an ML model quickly on the fly with little data or supervision? **That sounds hard.**"

"Ah, that's the point of the talk..."

"This speaker has something to say, maybe I shouldn't check my phone."

Now that it's framed...

...related work can now be discussed in the context of this framing

Establishing framing is the primary value of the talk intro

In this example: how have other folks tried to quickly adapt ML models to new contexts?

The "meta learning" field tries to train models that can be quickly adapted later using a few examples...

There's a whole body of work on "domain adaptation": taking a model that works well in context X and modifying it to work well in Y...

Instead, we adopt an approach based on "model distillation"

Tip III

Inputs, outputs,
constraints

Establish goals early

“Given these **inputs**, we wish to generate these **outputs**...”

“We are working under the following **constraints**”

Example: the outputs should have these properties...

Example: the computer graphics algorithm...

- Should run in real time

- Should be widely parallelizable so it can run efficiently on a GPU

Example: the system...

- Need not compile all of Python, only this subset that we care about...

- Should realize about 90% of the performance of hand-tuned code

Why goals and constraints?

Your contribution is typically a system or algorithm that meets the
stated goals under the stated constraints

Understanding whether a solution is “good” requires having this
problem context.

Tip IV

Show, don't tell

Example

In a recent project, Kayvon Fatahalian and team asked the question: given enough video of tennis matches of a professional athlete, could we turn it into a controllable video game character?

Words are ambiguous. Consider possible listener questions?

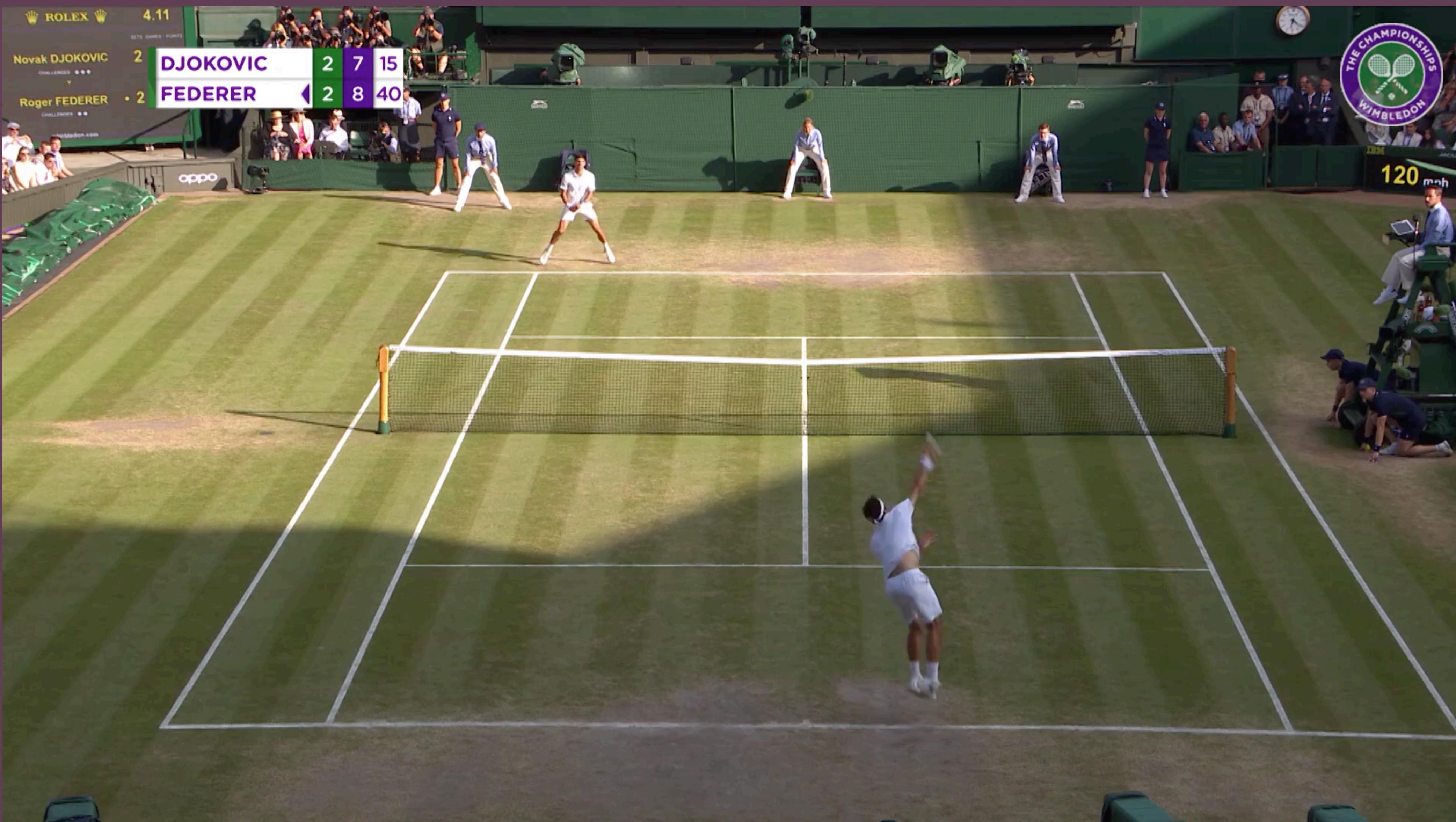
What does this video contain? Is it easy to analyze?

What do you mean by controllable characters?

What output are you trying to create? Video? 3D animation?

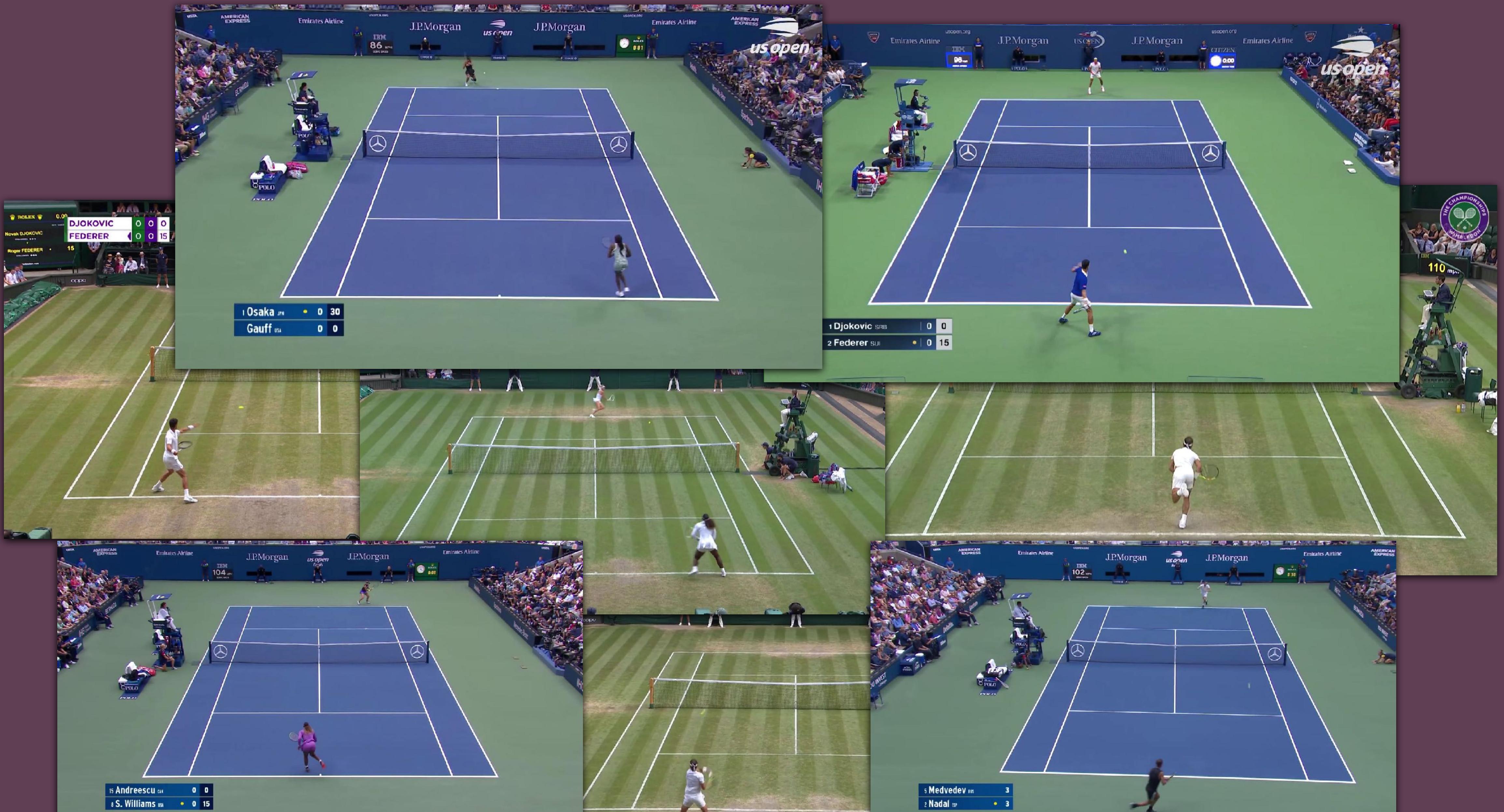
Compare the description above to the following sequence...

Example source video

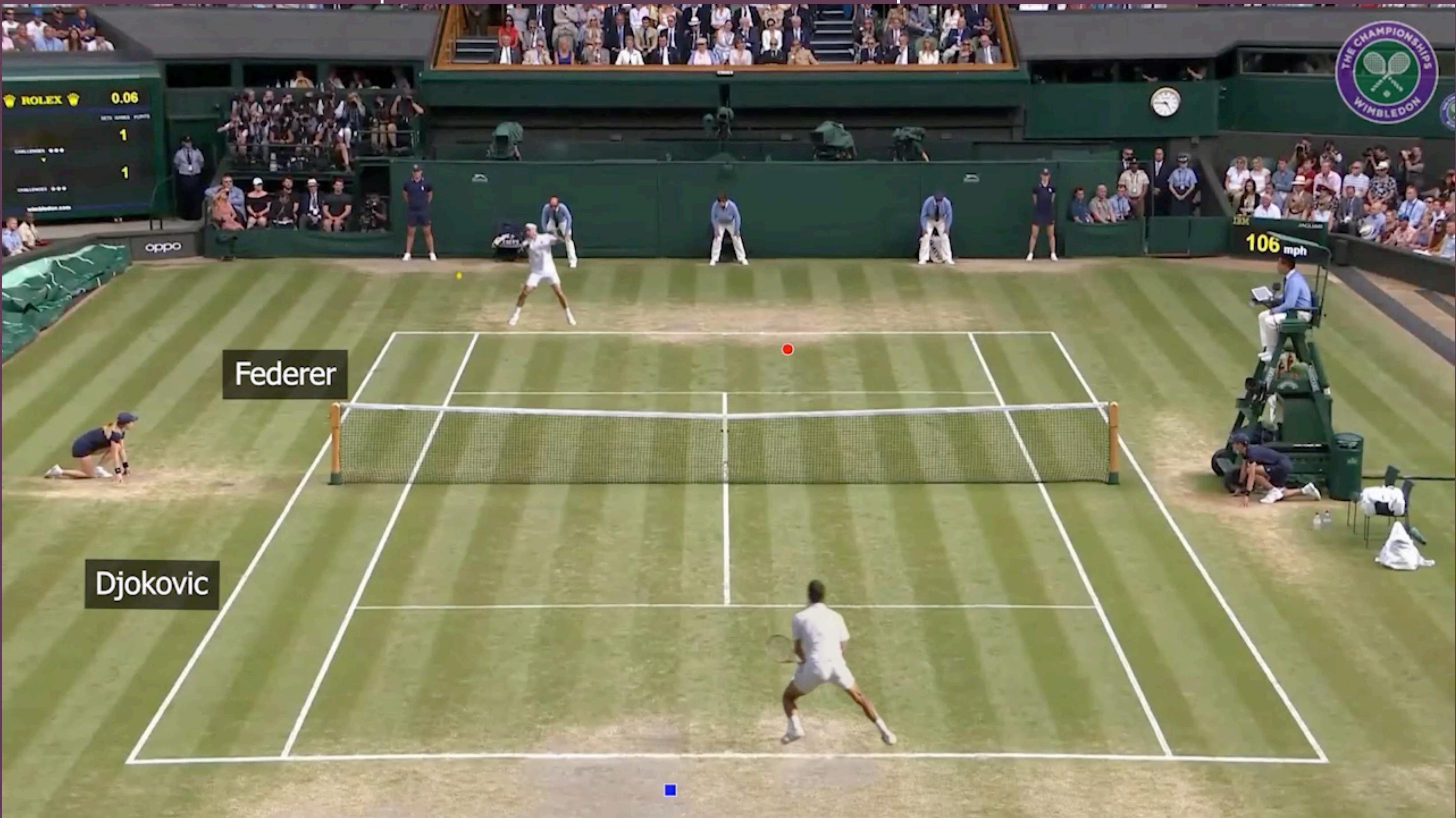


The best way to describe the input data is just show it!

And there's a lot of it!

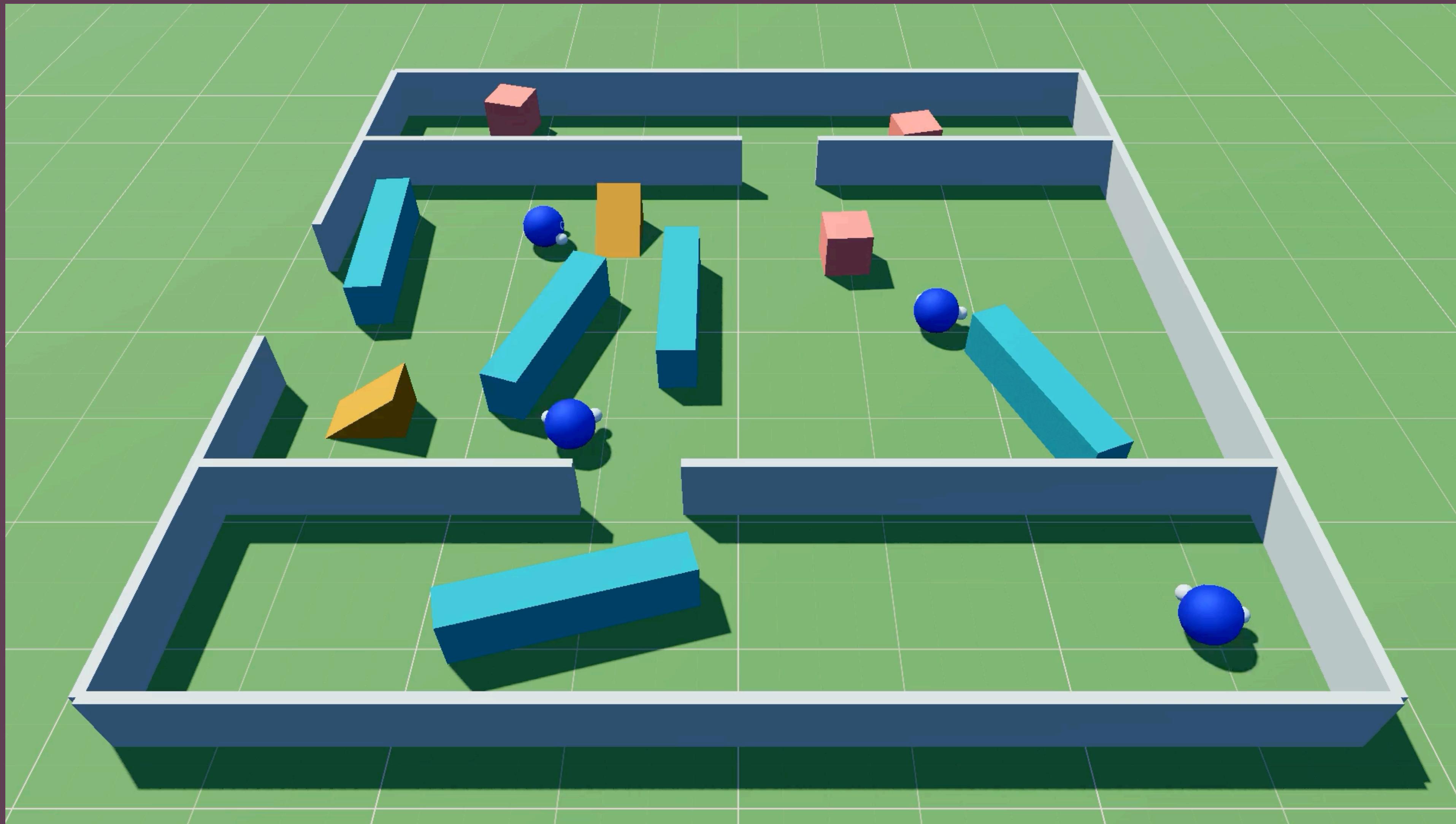


Here's an example of controllable output



The best way to describe the output we seek is just show the result of the system

Another example

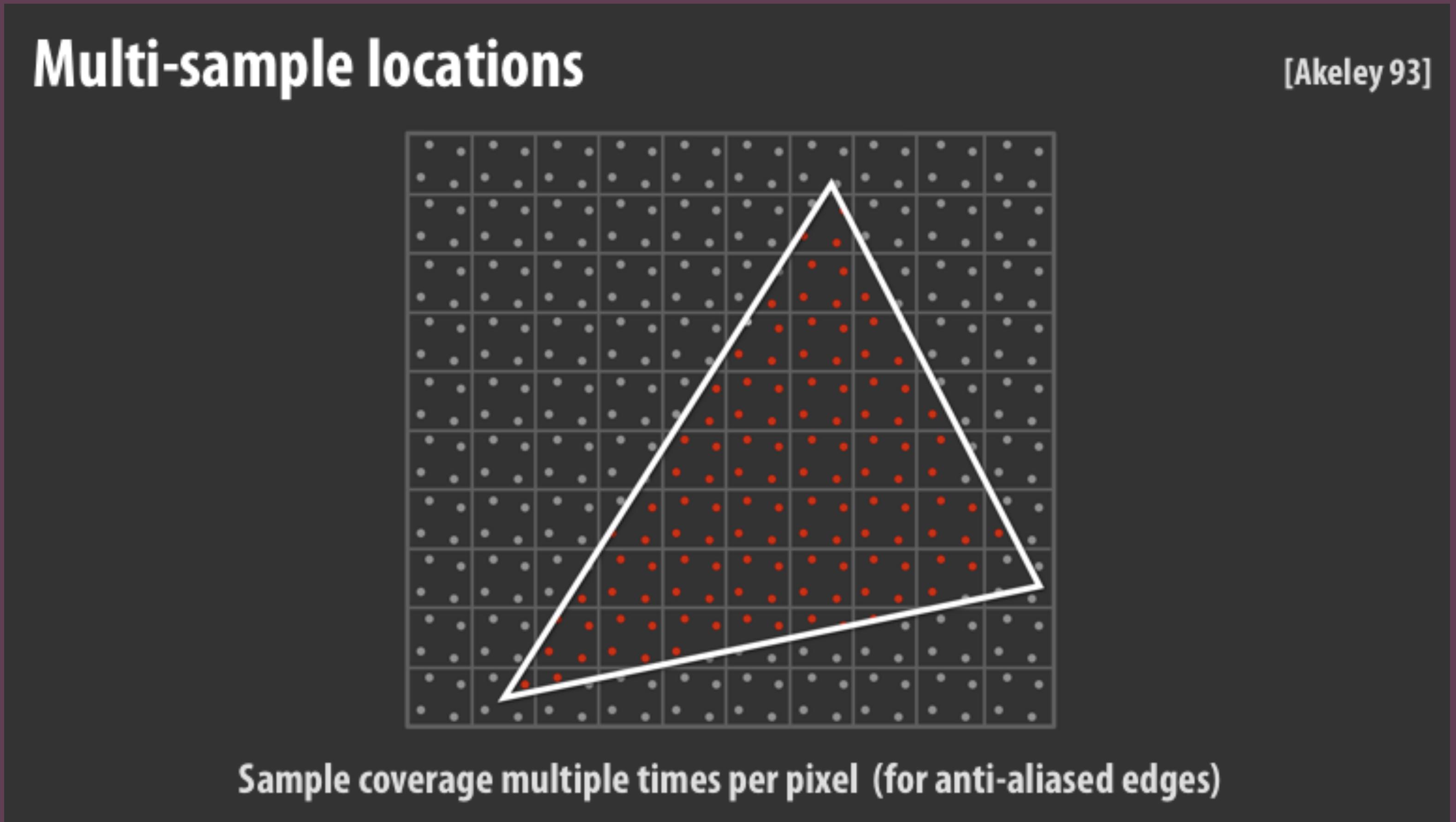


A game engine
that can play
many games in
parallel
simultaneously

Tip V

Always explain any
figure or graph

Walk the audience through any figure

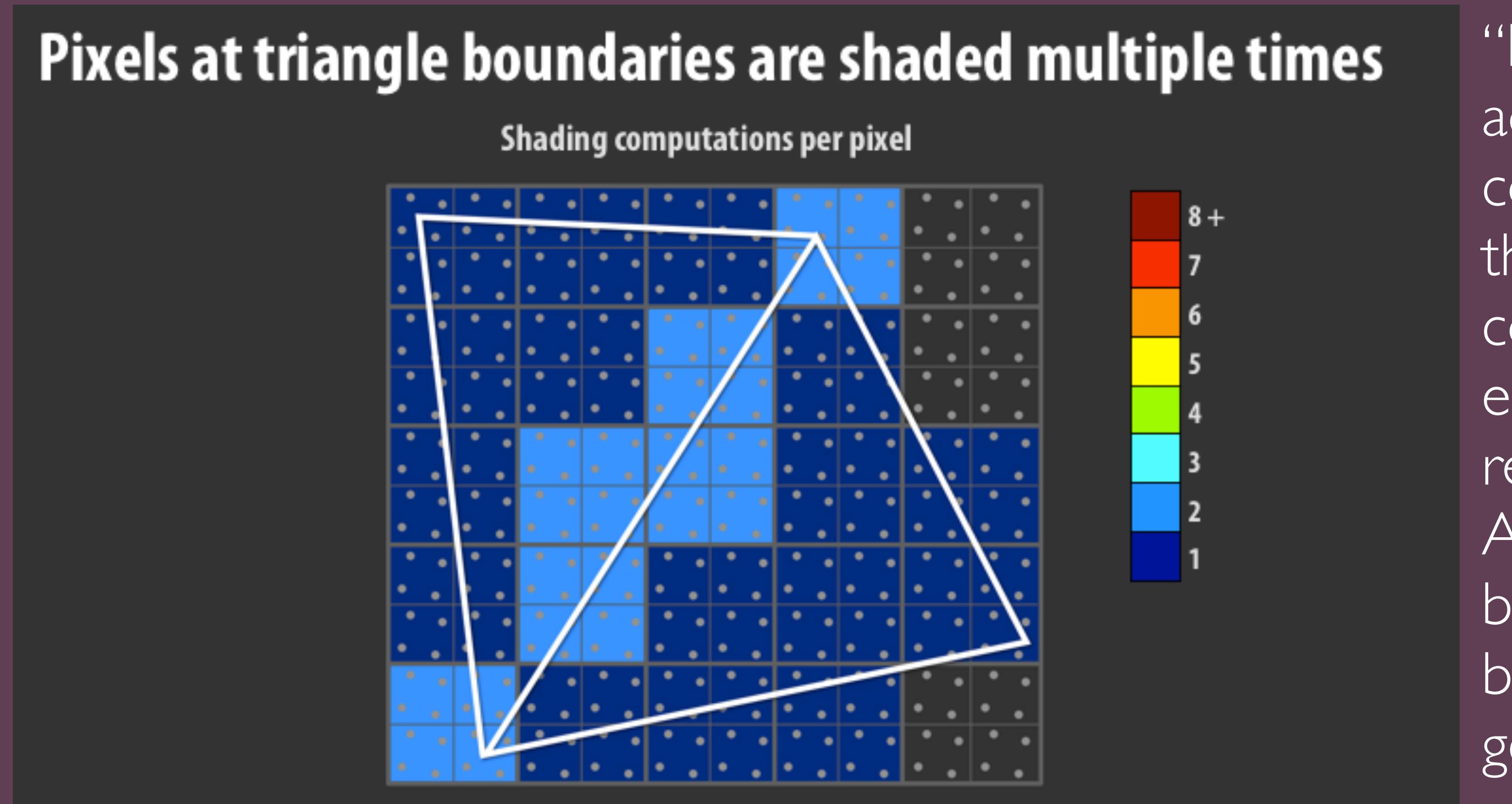


What are we looking at?

Looking at this, you don't know what it is or why I'm showing it to you

How about: "Here I'm showing you a pixel grid, a projected triangle, and the location of four sample points at each pixel. Sample points falling within the triangle are colored red."

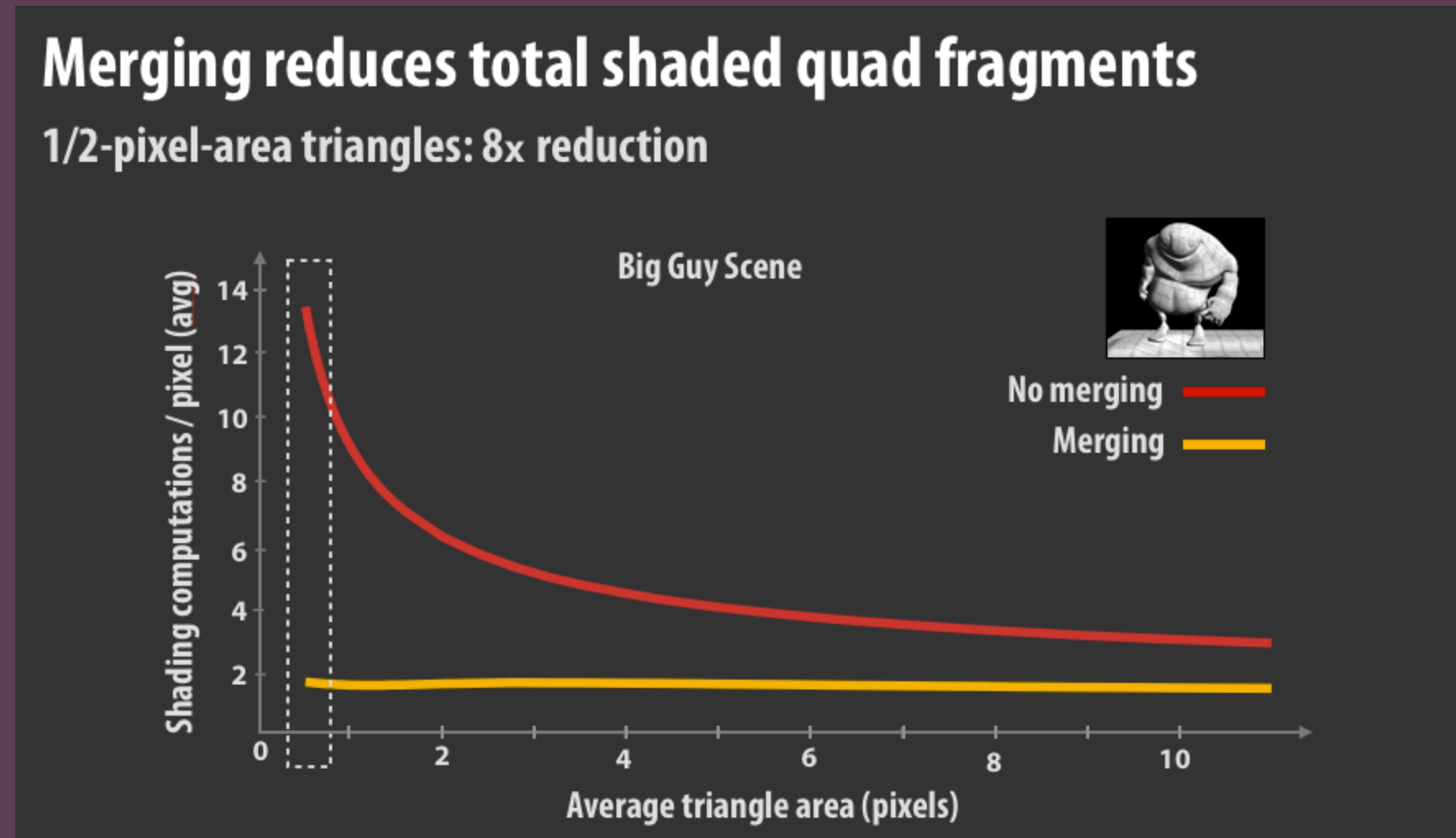
Walk the audience through any figure



What should I take away?

“Now I’m showing you two adjacent triangles, and I’m coloring pixels according to the number of shading computations that occur at each pixel as a result of rendering these two triangles. As you can see from the light blue region, pixels near the boundary of the two triangles get shaded twice.”

Walk the audience through any figure



Explain the axes and result I should notice

"This figure plots the number of shading computations when rendering tessellations. X-axis gives triangle size. If you look at the left side of the graph, which corresponds to a high-resolution micropolygon mesh, you can see that merging, shown by yellow line, shades over eight times fewer per pixel."

Tip VI

One point per slide

One point per slide

One point per slide

One point per slide

Don't do this:

Results

I. Quantitative

I. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec metus dolor, accumsan feugiat porta a, placerat ut risus. Curabitur eu sapien

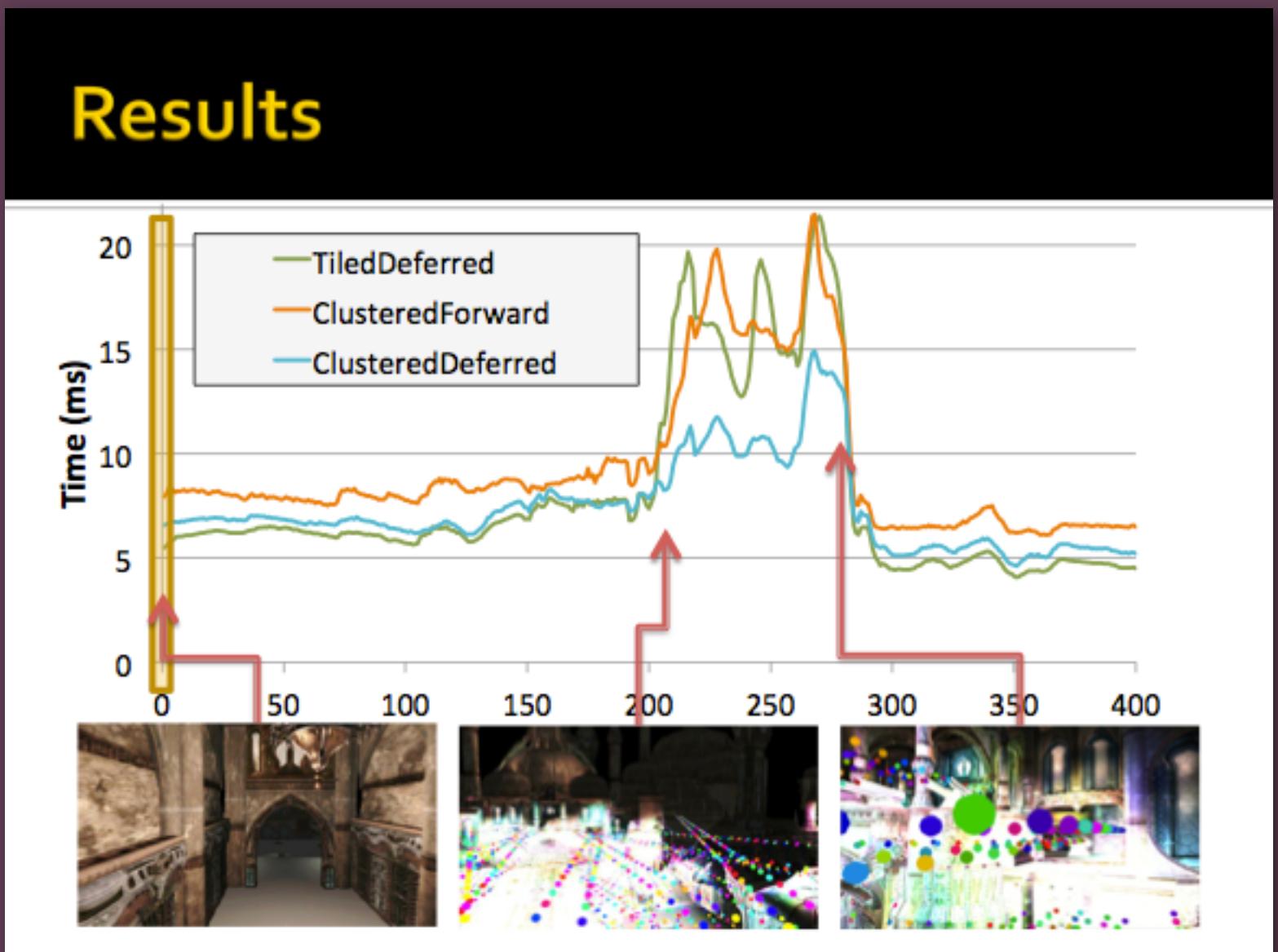
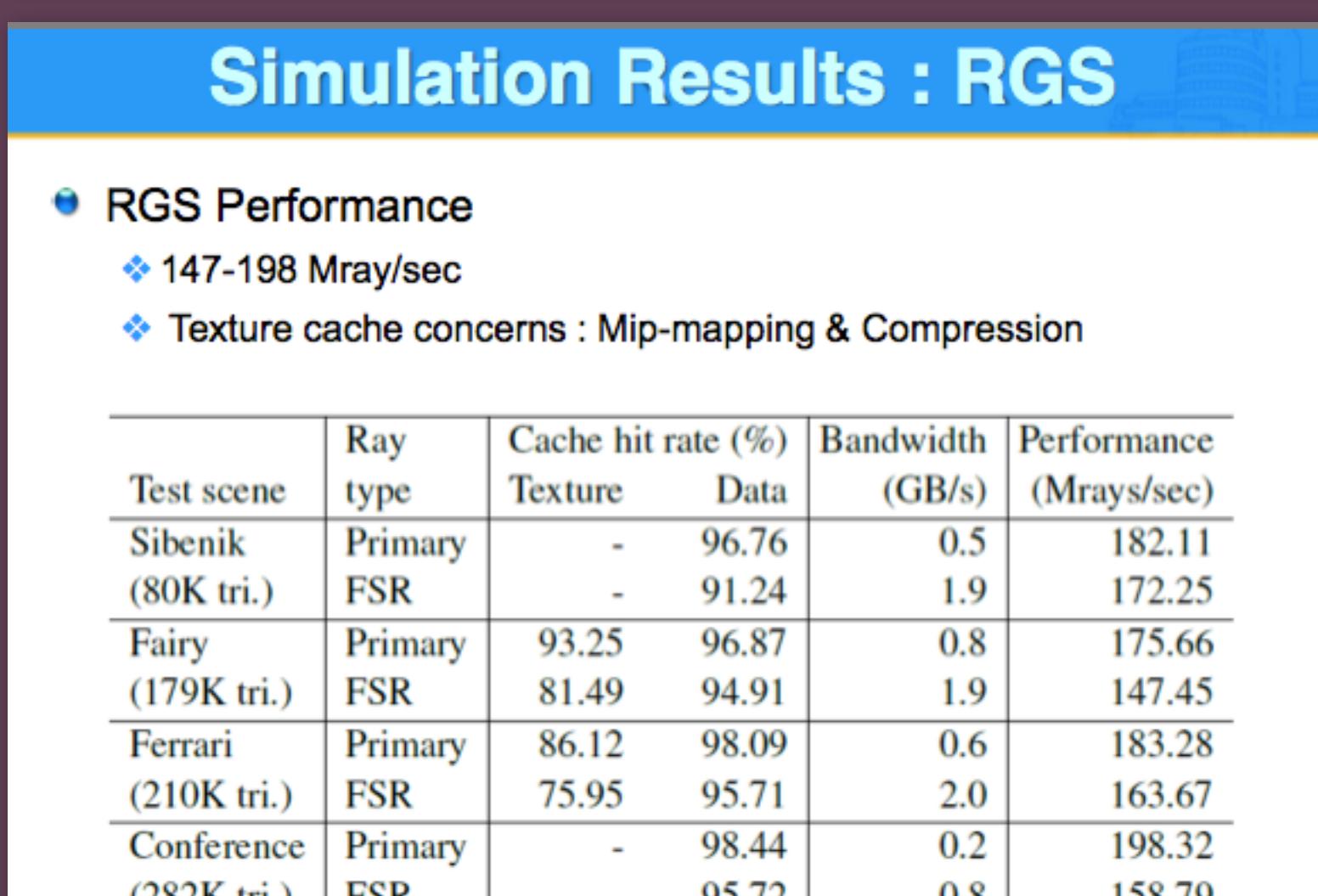
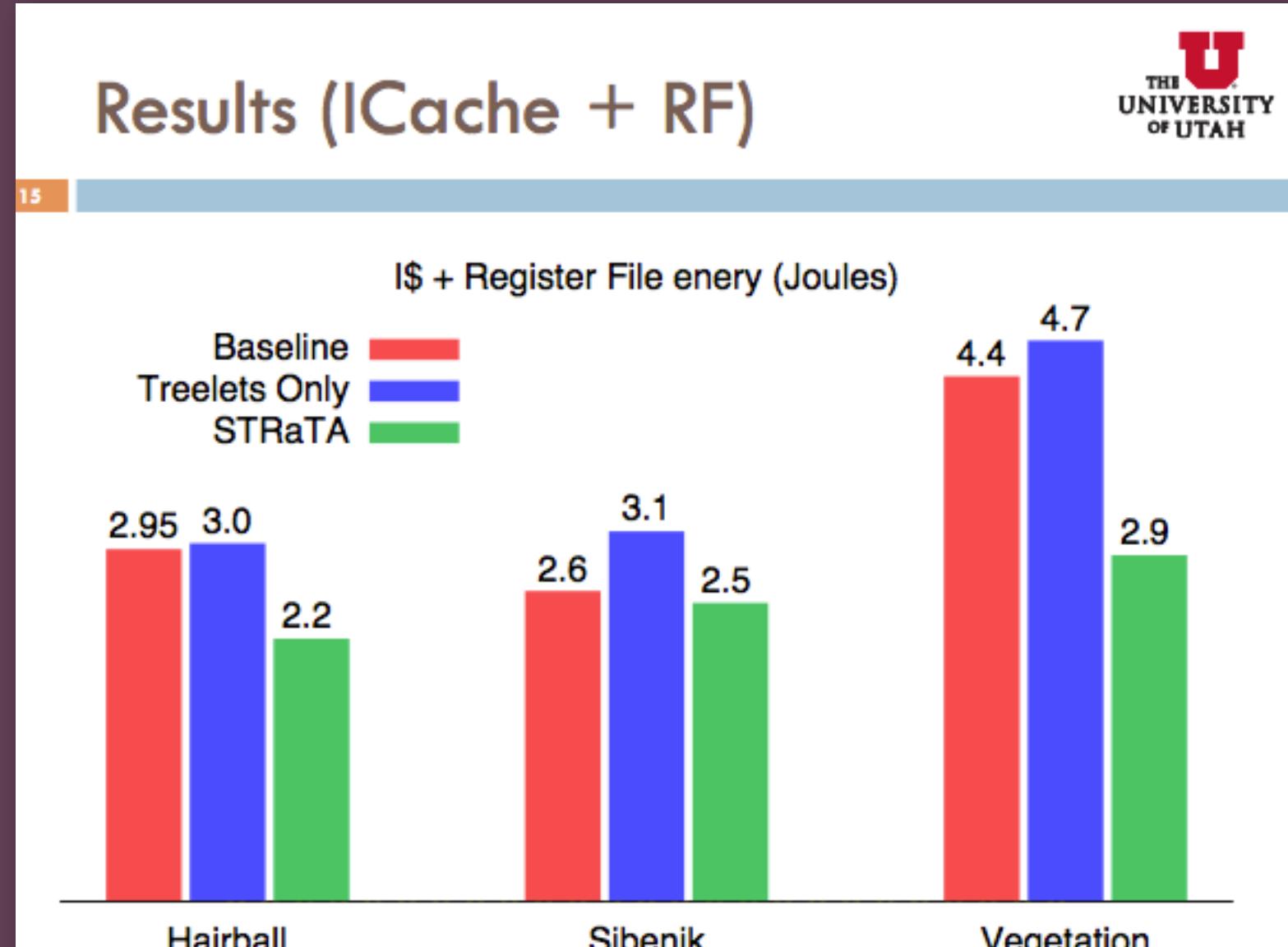
2. Qualitative

I. Integer sit amet justo risus. Nunc at facilisis urna. Pellentesque cursus et lorem sed fringilla. Etiam tincidunt tortor at tincidunt iaculis. Maecenas

3. Performance

I. Nullam odio turpis, ultricies a arcu id, feugiat pretium sem. Nullam arcu erat, faucibus id laoreet ullamcorper, lacinia vitae ligula.

Bad examples



You have to work very hard to interpret these data—what are you supposed to take away?

Steve Krug

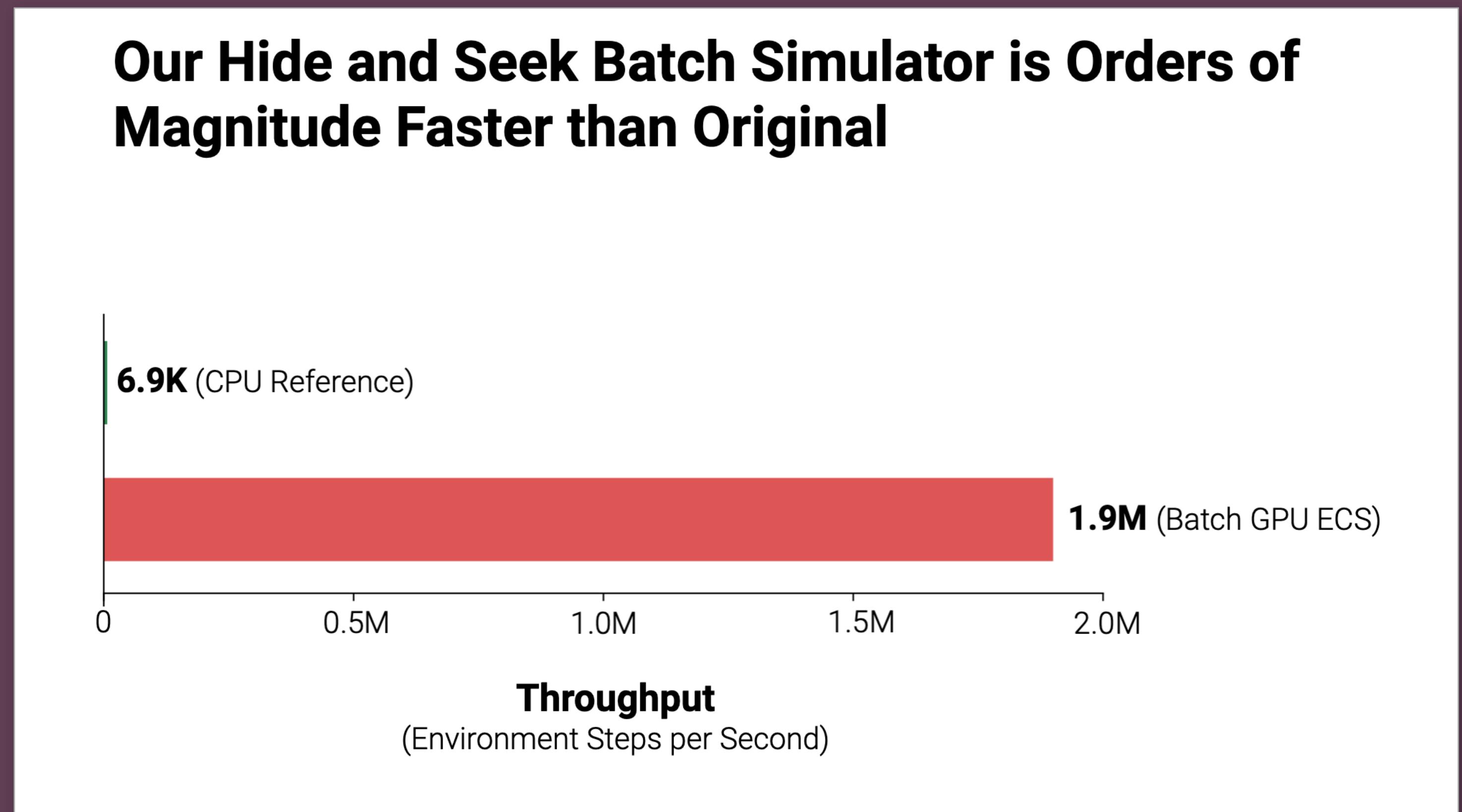


DON'T
MAKE
ME
THINK

www.krug.com

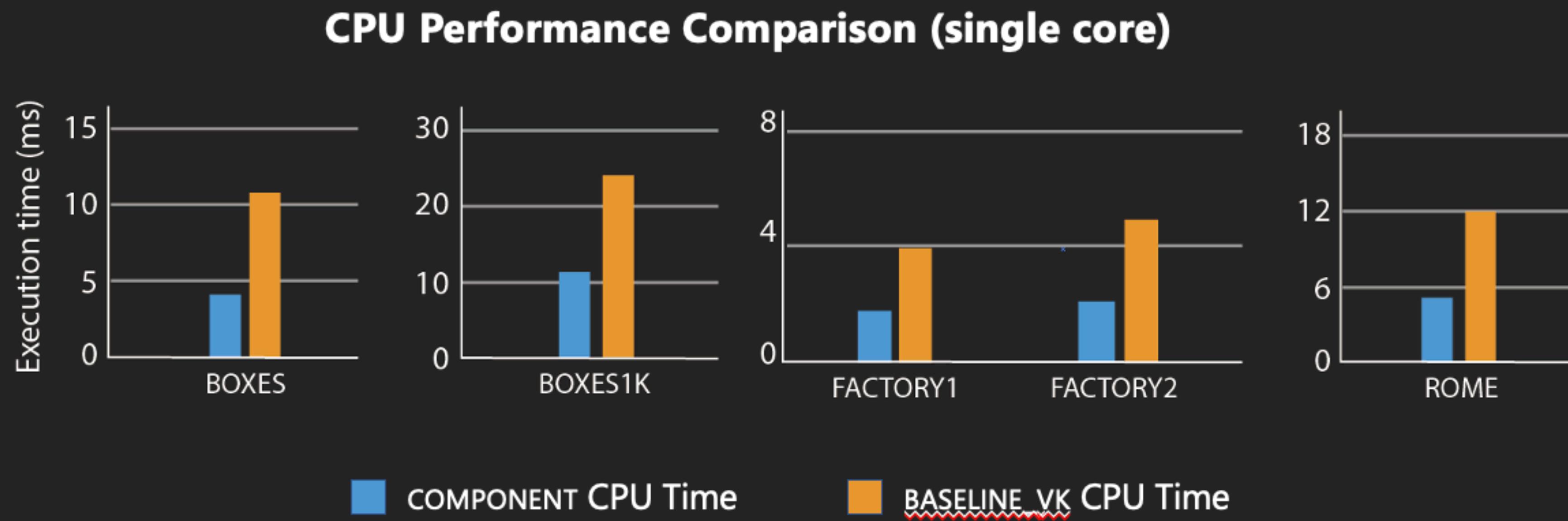
The point is the slide title.

Every slide should have a thesis you are proving on that slide, and that thesis should be the title of the slide

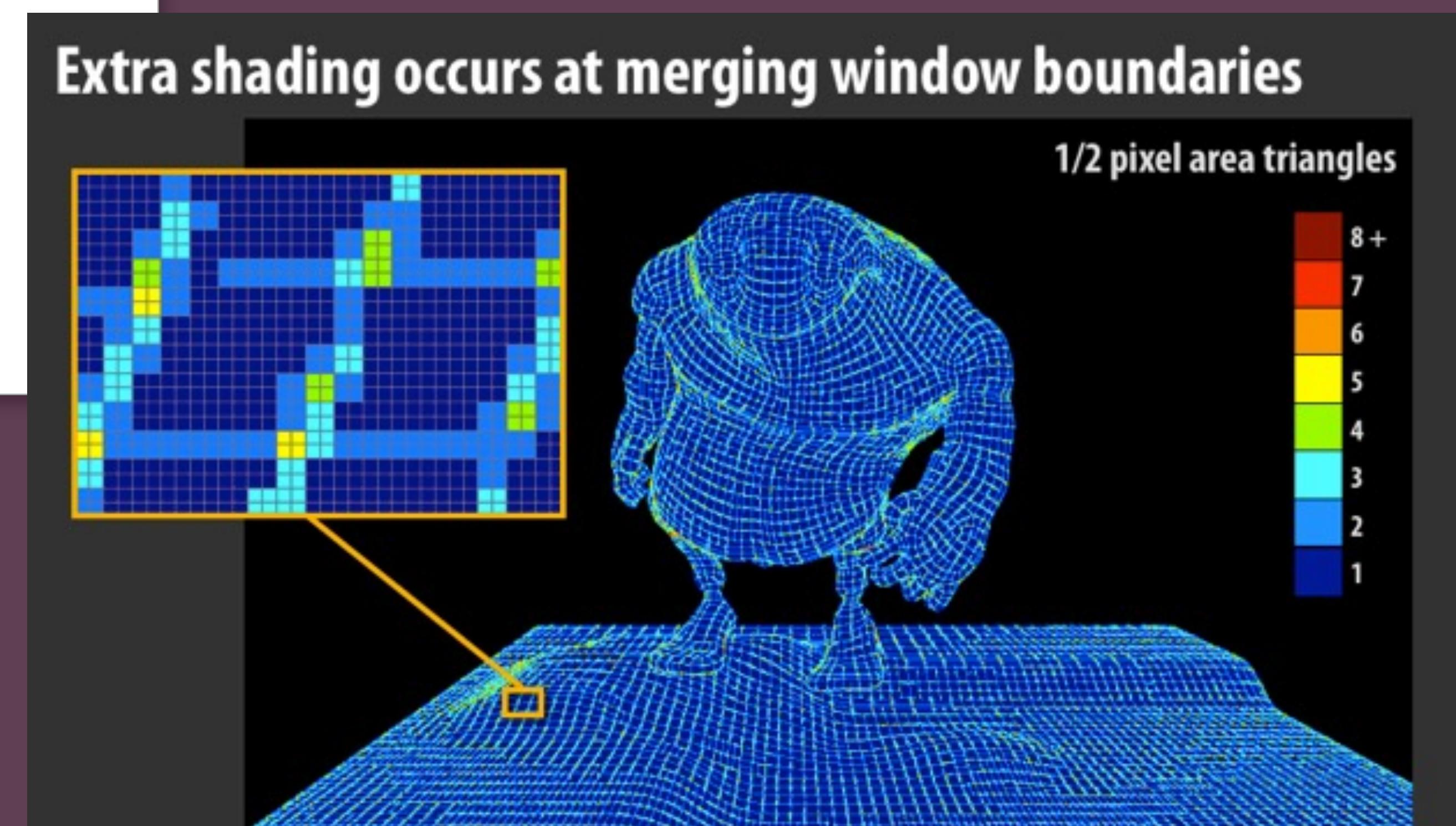
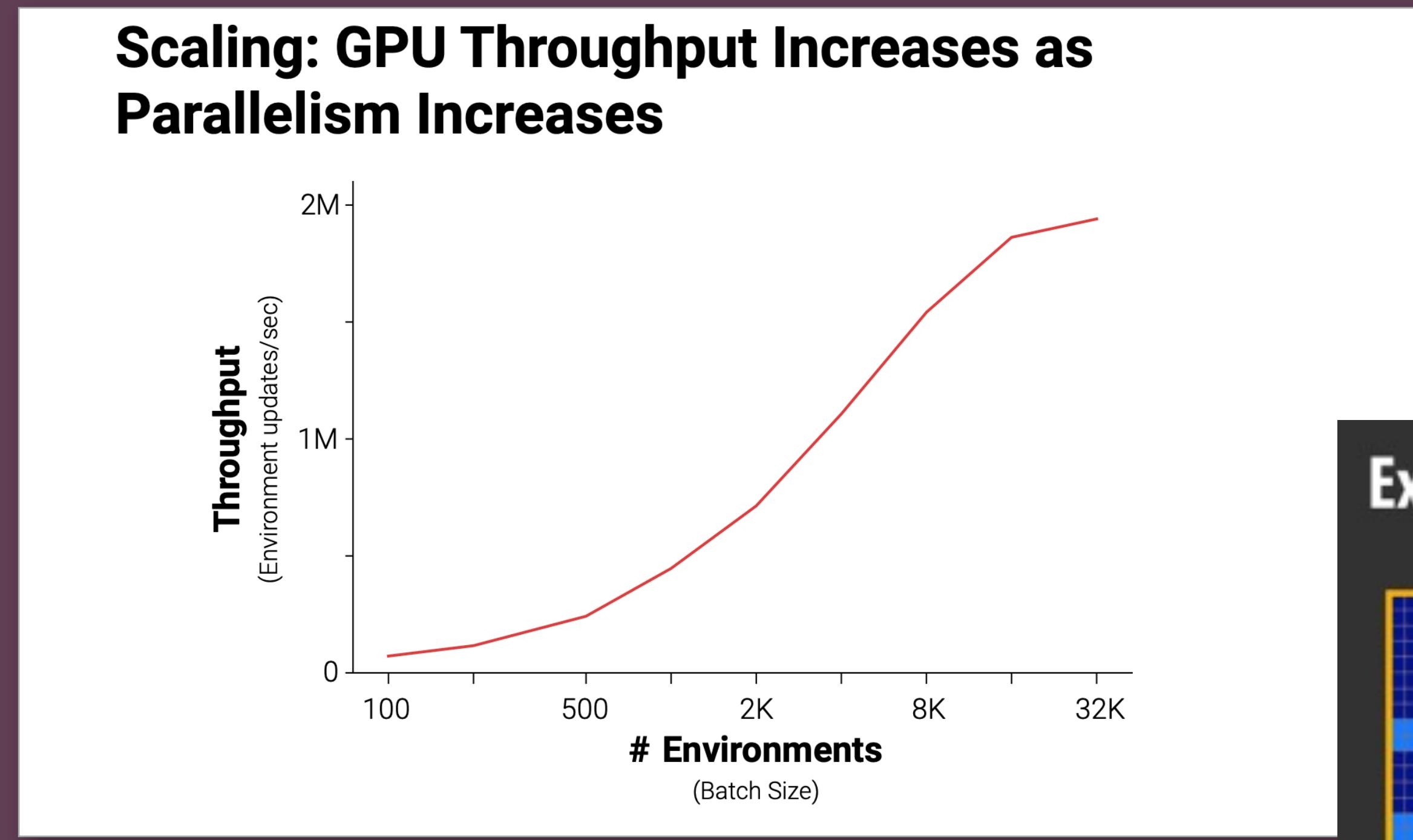


Example

The **COMPONENTS** renderer uses **2x less CPU time than BASELINE_VK**



More examples



Slide titles matter!

If you read the titles of your talk all the way through, it should be a great summary of the talk.

Tennis has a cyclic structure

The diagram illustrates a cyclic process with three stages: "Ready" (Phase = 0/2π), "Reaction" (a black square), and "Phase = π". Arrows indicate a clockwise flow between the stages.

GPUs shade quad fragments (2x2 pixel blocks)

The diagram shows "Texture data" (a yellow checkered pattern) and a "Quad fragment" (a 2x2 grid). A red arrow labeled $\frac{ds}{dx}$ points from the top-left corner of the quad fragment to a point on the texture data, illustrating how derivatives are estimated from neighboring texel coordinates.

AAC IS AN APPROXIMATION TO THE TRUE AGGLOMERATIVE CLUSTERING SOLUTION.

Computation graph:

A computation graph with several nodes represented by blue squares and one large central node containing four smaller blue squares. Edges connect the nodes.

Primitive partitioning:

A diagram showing a grid of triangles, likely representing a mesh or a set of primitives being partitioned.

use differences between neighboring texture coordinates to estimate derivatives

Read your talk as thumbnails

1. Form-From: A Design Space of Social Media Systems. A 2x2 matrix with 'Flat' and 'Threaded' on the top and 'Spaces' and 'Network' on the right.

2. The more things change...: A slide showing various social media icons (X, M, Notes, etc.) and asking what makes them different.

3. Instagram vs. TikTok feed comparison: A slide comparing Instagram's infinite scroll with TikTok's platform-wide feed.

4. Design spaces: Johansen's (1988) Time-Space Matrix. A 2x2 matrix with 'Same time' and 'Different time' on the top and 'Same place' and 'Different place' on the right.

5. Design spaces: Johansen's (1988) Time-Space Matrix. A detailed version of the matrix with specific platform examples.

6. Our goal: Introducing a design space of social media systems that captures modern similarities and differences.

7. Method: Inductive analysis via purposive sampling of >100 public platforms and social computing research systems. Shows 62-Dimension Full Model, 11-Category Model, and 2-Category Form-From Model.

8. Form-From: A 2x2 matrix with 'Flat' and 'Threaded' on the top and 'Spaces', 'Network', and 'Commons' on the right.

9. Form that content takes: Flat vs Threaded. Shows examples of flat streams and threaded replies.

10. From where content comes: Flat, Threaded, Spaces, Network, Commons. Shows examples of joinable locations, friend/follow relationships, and drawn-from-the-entire-platform.

11. Flat vs Threaded: Posts are explicitly linked to each other. Shows examples of Slack, Discord, Clubhouse, IRC, and others.

12. Flat vs Threaded: Posts are not explicitly linked. Shows examples of Reddit, Email, Quora, and others.

13. Look for design patterns within cells of Form-From: Threaded Spaces. Shows examples of phpBB and bbPress.

14. Look for design patterns within cells of Form-From: Threaded Network. Shows examples of Twitter and LinkedIn.

15. Look for design patterns within cells of Form-From: Threaded Commons. Shows examples of Wikipedia and GitHub.

16. Evolution over time: Timeline of major social media platforms. Shows the evolution from the first online chat rooms to the rise of social media platforms like Facebook and YouTube.

Do they tell
the story
that you
want to tell?

(Side benefit:
assume
people can't
read anything
too small in
this view)

In sum:

1. Identify your one narrative goal
2. Every sentence matters
3. Inputs, outputs, constraints
4. Show, don't tell
5. Always explain any figure or graph
6. One point per slide

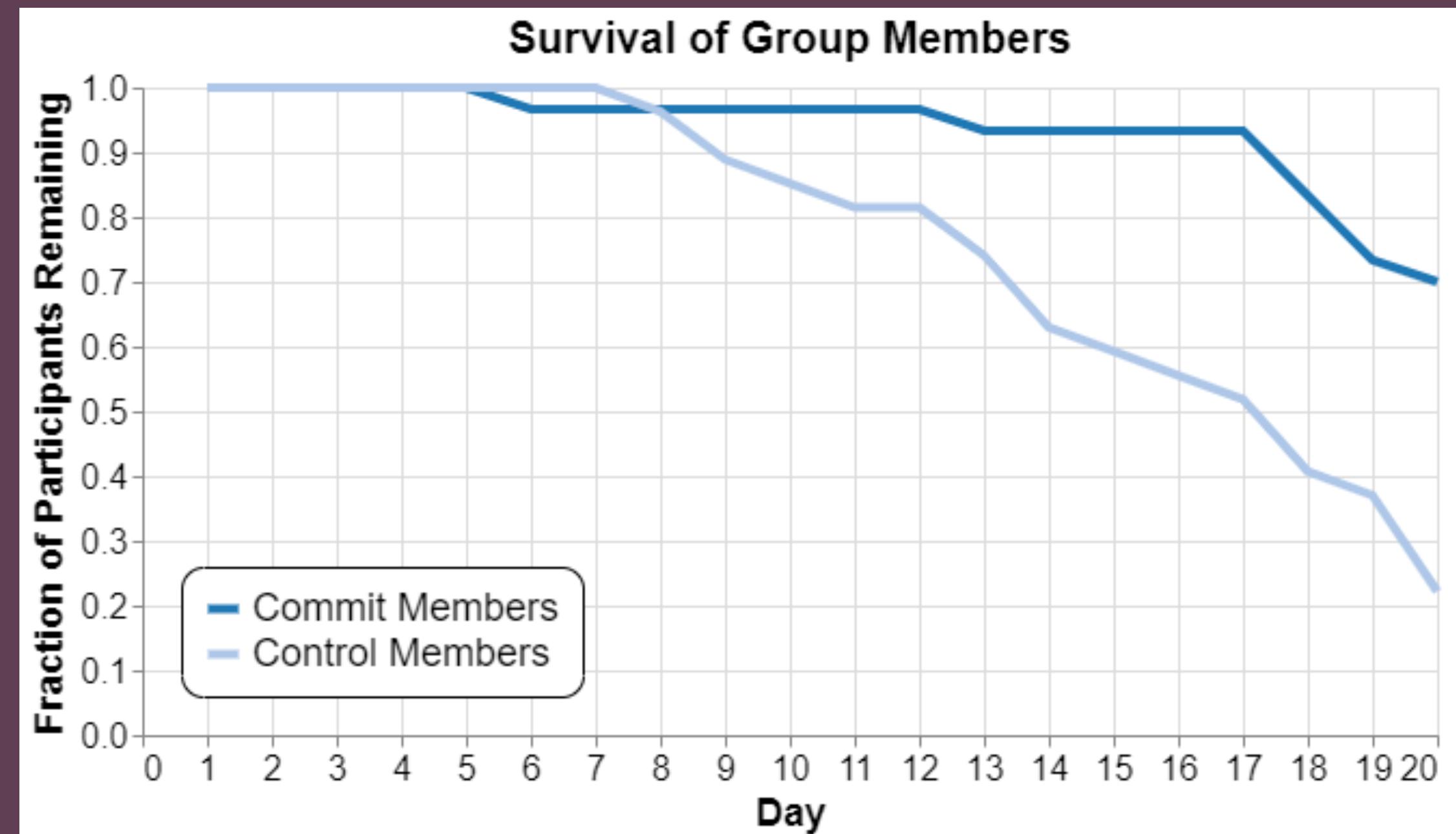
Figures

Each figure makes one point

The first sentence of the caption of the figure should describe what the person takes away from the figure, not what is in the figure

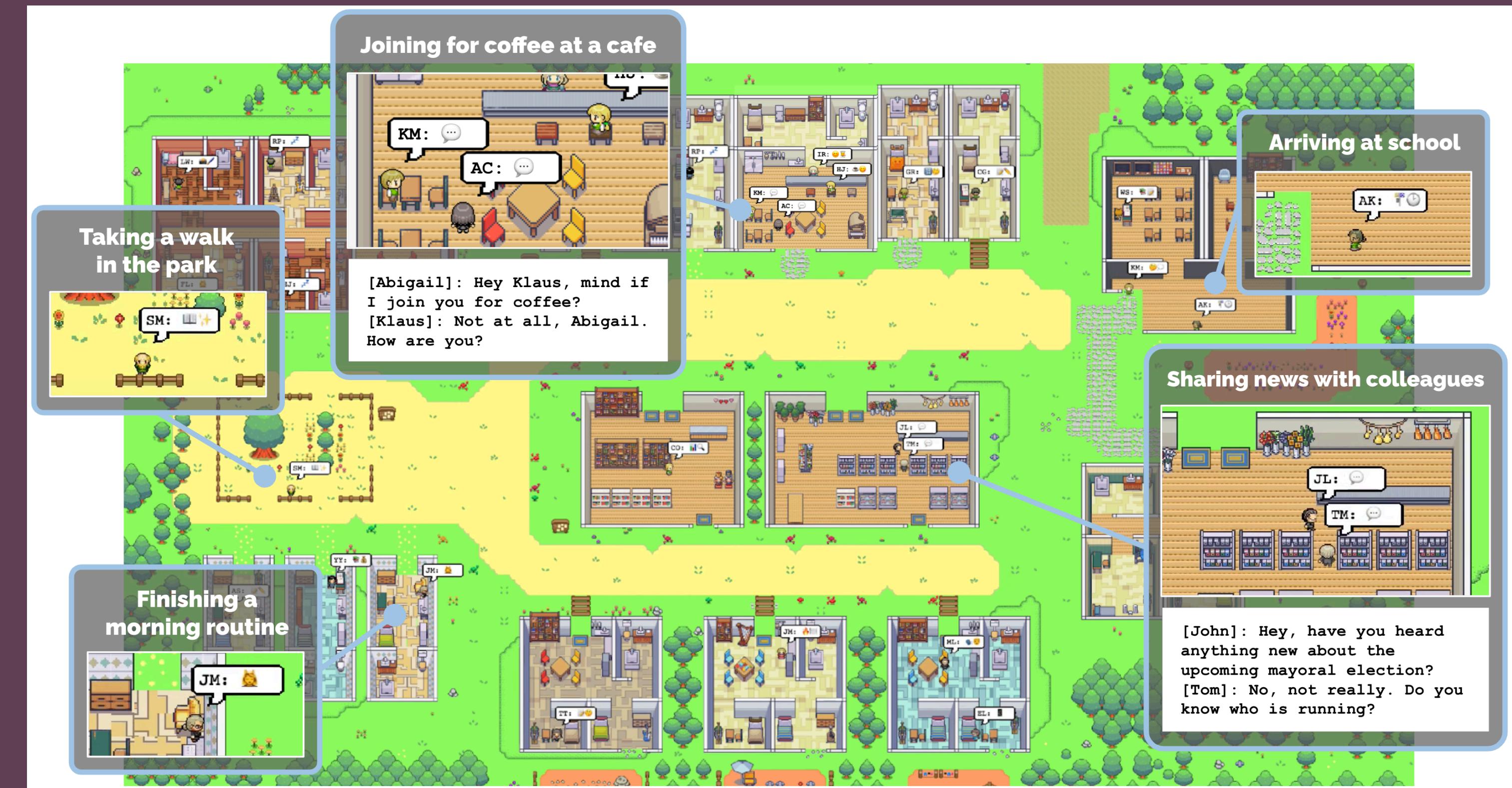
Not: “A survival plot comparing the Commit to the Control condition.”

Instead: “Participants in the Commit condition were four times as likely to stay active until the end of the study.”



Each figure makes one point

Ergo, ask yourself: **what is the point being made by this figure?**
What justifies its existence?



“We made a simulated town”

Common figure types

Introductory / overview of what you did

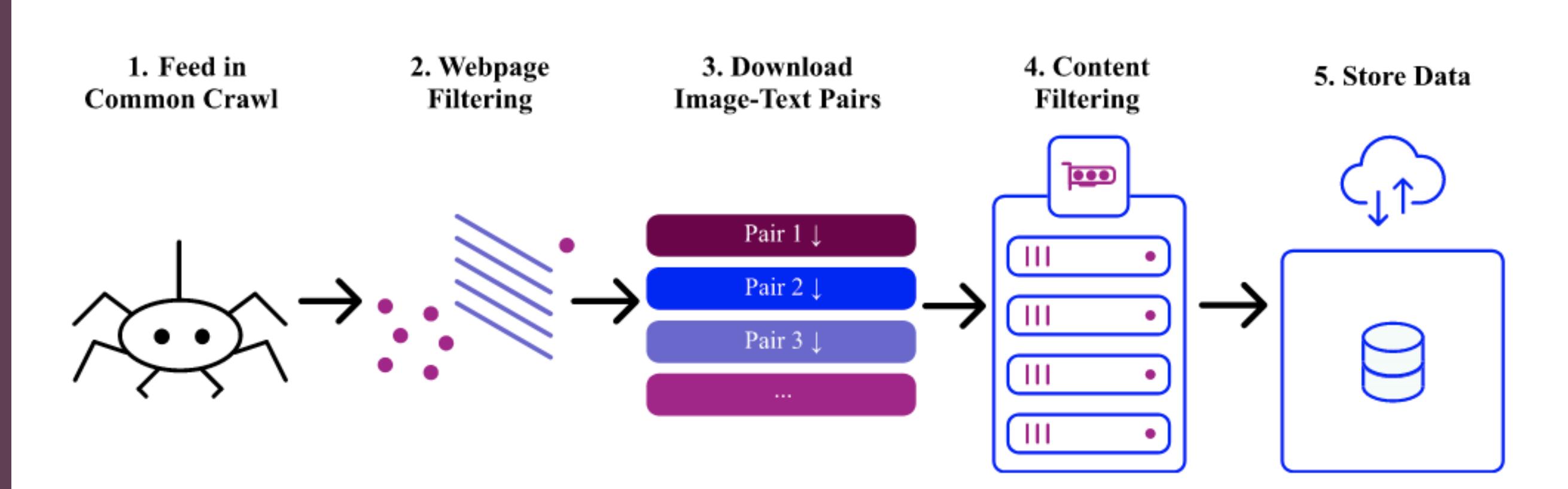
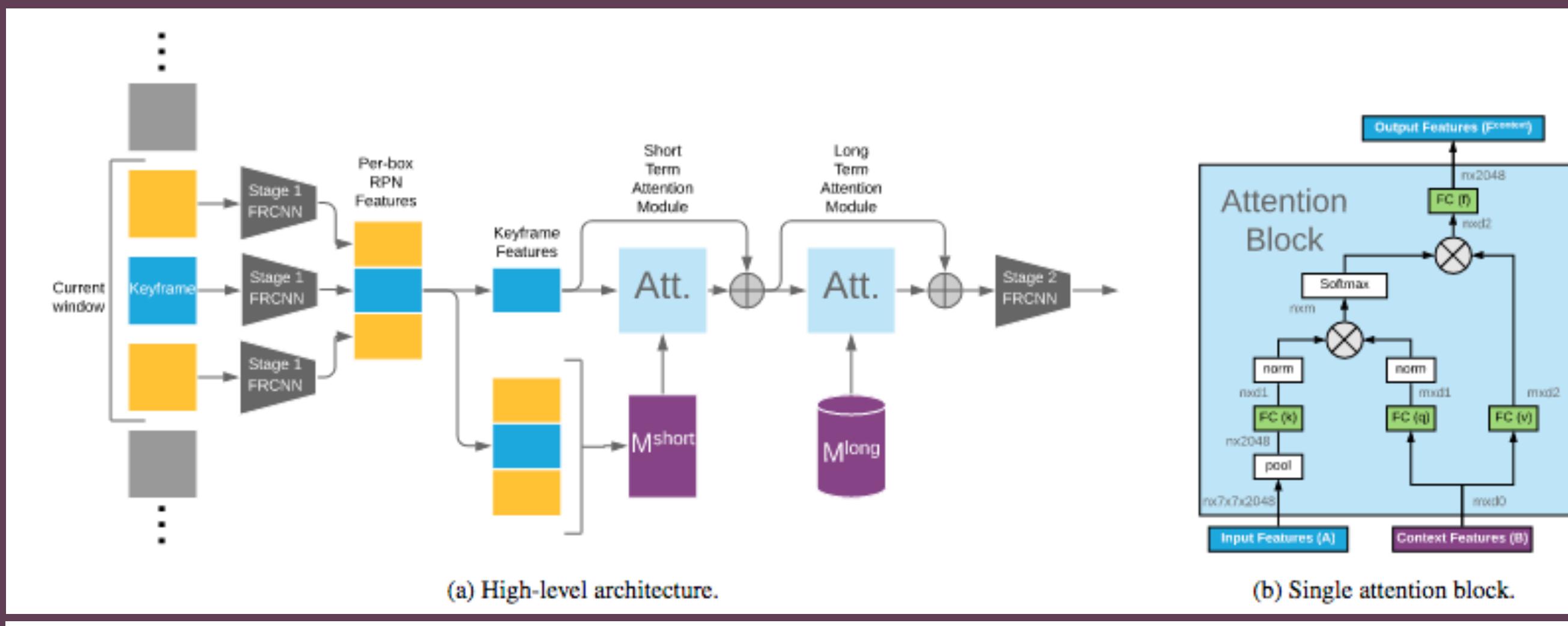


Figure 1: Visual similarity over long time horizons. In static cameras, there exists significantly more long term temporal consistency than in data from moving cameras. In each case above, the images were taken on separate days, yet look strikingly similar.

Beery et. al. 2020

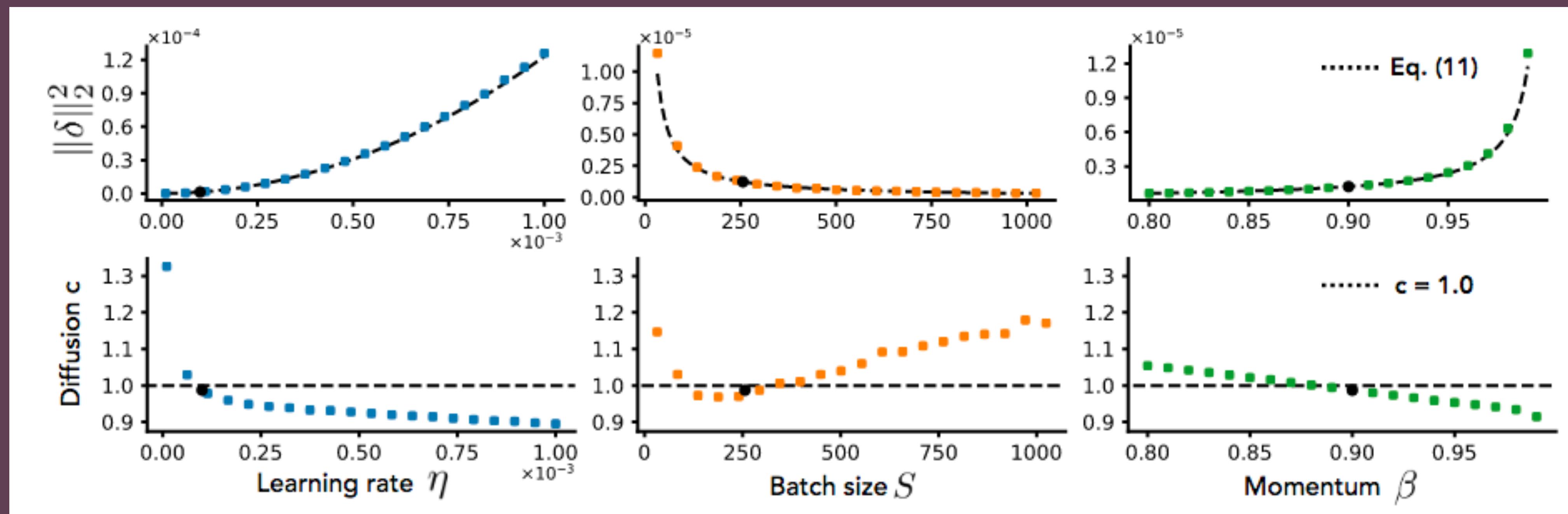
Common figure types

Framework / pipeline / model



Common figure types

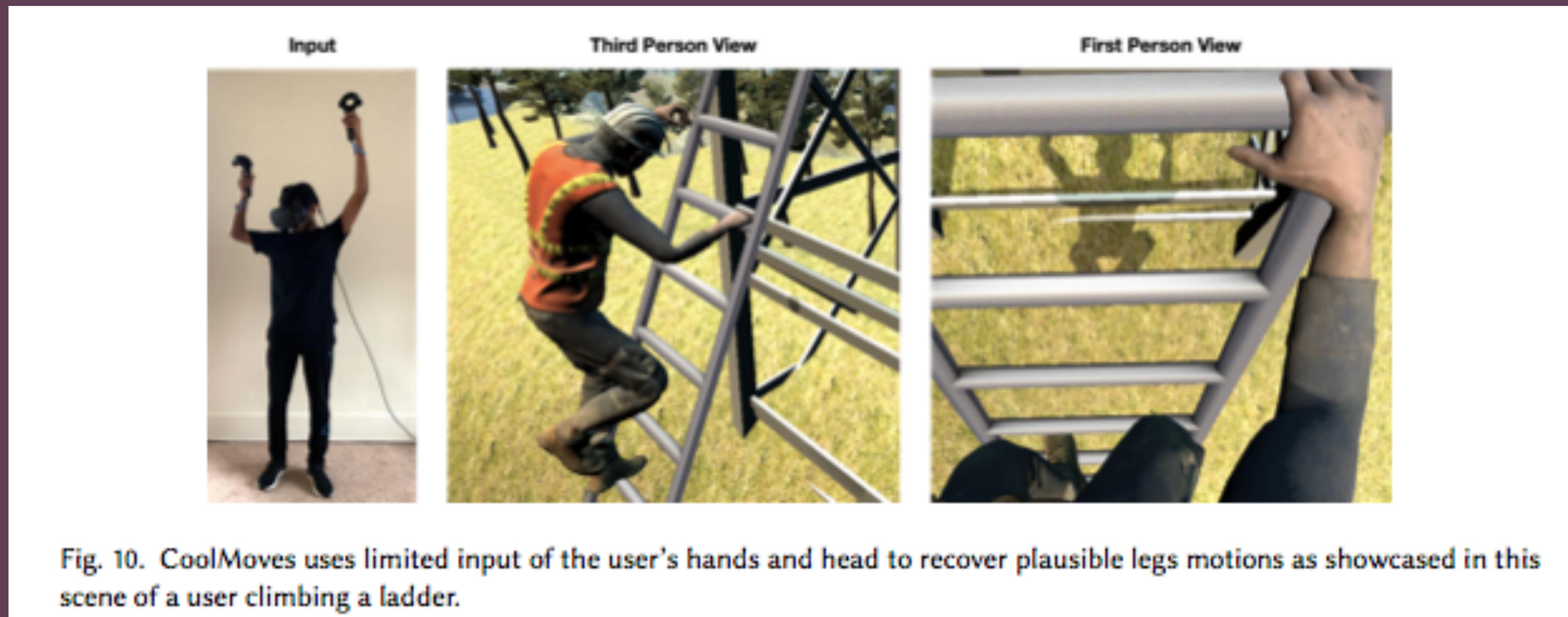
Experimental results



Kunin et. al. 2020

Common figure types

Applications or use in practice



Ahuja et. al. 2021

What makes a good figure?

A good figure does:

Summarize one key takeaway at a time

Make it immediate and easy to see your bit flip

Remain true to the data

A good figure does **not**:

Become a giant information dump of all your findings

Bury your bit flip in a hard-to-see format

Hide mediocre results with questionable display

Tuning your tables

Tables are the neutron stars of your paper: incredibly information-dense

Ideal for summarizing 3+ treatments

Centering your bit flip:

Can **bold** best model per-treatment

Clever alternatives: arrows, colored
deltas

Use colors, lines, bolding to distinguish
treatments & improve readability

Model	Pre-training	INet	INet-v2	INet-R	INet-S	ObjNet
B/32	CLIP WIT	63.3	56.0	69.4	42.3	44.2
	LAION-400M	62.9 ^{-0.4}	55.1 ^{-0.9}	73.4 ^{+4.0}	49.4 ^{+7.1}	43.9 ^{-0.3}
	LAION-2B-en	65.7 ^{+2.4}	57.4 ^{+1.4}	75.9 ^{+6.5}	52.9 ^{+10.6}	48.7 ^{+4.5}
B/16	CLIP WIT	68.3	61.9	77.7	48.2	55.3
	LAION-400M	67.0 ^{-1.3}	59.6 ^{-2.3}	77.9 ^{+0.2}	52.4 ^{+4.2}	51.5 ^{-3.8}
B/16+	LAION-400M	69.2	61.5	80.5	54.4	53.9
L/14	CLIP WIT	75.6	69.8	87.9	59.6	69.0
	LAION-400M	72.8 ^{-2.8}	65.4 ^{-4.4}	84.7 ^{-3.2}	59.6	59.9 ^{-9.1}
	LAION-2B-en	75.2 ^{-0.3}	67.7 ^{-2.0}	87.4 ^{-0.5}	63.3 ^{+3.7}	65.5 ^{-3.6}

Schuhmann et. al. 2022

Read your paper as figures

Just like the talk thumbnail view, you should be able to understand the main narrative of the paper by just skimming the figures and tables

Generative Agents: Interactive Simulacra of Human Behavior



Figure 1: Generative agents are believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of The Sims, with twenty-five agents. Users can observe and intervene as agents plan their days, share news, form relationships, and coordinate group activities.

ABSTRACT

Believable proxies of human behavior can empower interactive applications ranging from immersive environments to rehearsal spaces for interpersonal communication to prototyping tools. In this paper, we introduce generative agents: computational software agents that simulate believable human behavior. Generative agents wake up, cook breakfast, and head to work; artists paint, while

permissions to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
UIST '23, October 29–November 1, 2023, San Francisco, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9386-1/23/10.
https://doi.org/10.1145/3586183.3606763

authors write; they form opinions, notice each other, and initiate conversations; they remember and reflect on days past as they plan the next day. To enable generative agents, we describe an architecture that extends a large language model to store a complete record of the agent's experiences using natural language, synthesize those memories over time into higher-level reflections, and retrieve them dynamically to plan behavior. We instantiate generative agents to populate an interactive sandbox environment inspired by The Sims, where end users can interact with a small town of twenty-five agents using natural language. In an evaluation, these generative agents produce believable individual and emergent social behaviors. For example, starting with only a single user-specified notion that one agent wants to throw a Valentine's Day party, the agents autonomously spread invitations to the party over the next two

Isabella: I'm still weighing my options, but I've been discussing the election with Sam Moore. What are your thoughts on him?
Tom: To be honest, I don't like Sam Moore. I think he's out of touch with the community and doesn't have our best interests at heart.

We note that the conversational style of these agents can feel overly formal, likely a

Generative Agents

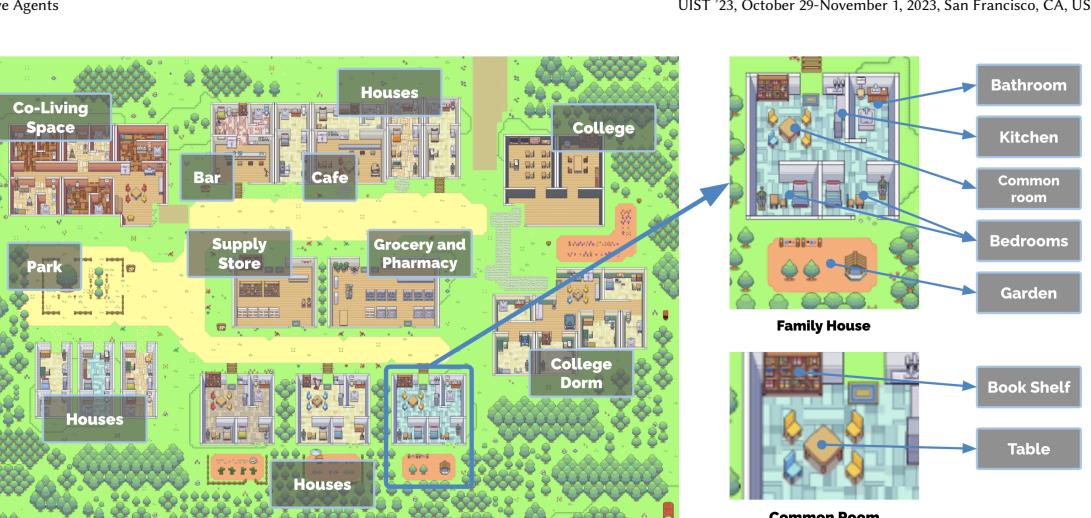


Figure 2: The Smallville sandbox world, with areas labeled. The root node describes the entire world, children describe areas (e.g., houses, cafe, stores), and leaf nodes describe objects (e.g., table, bookshelf). Agents remember a subgraph that reflects the parts of the world they have seen, maintaining the state of those parts as they observed them.

Generative Agents

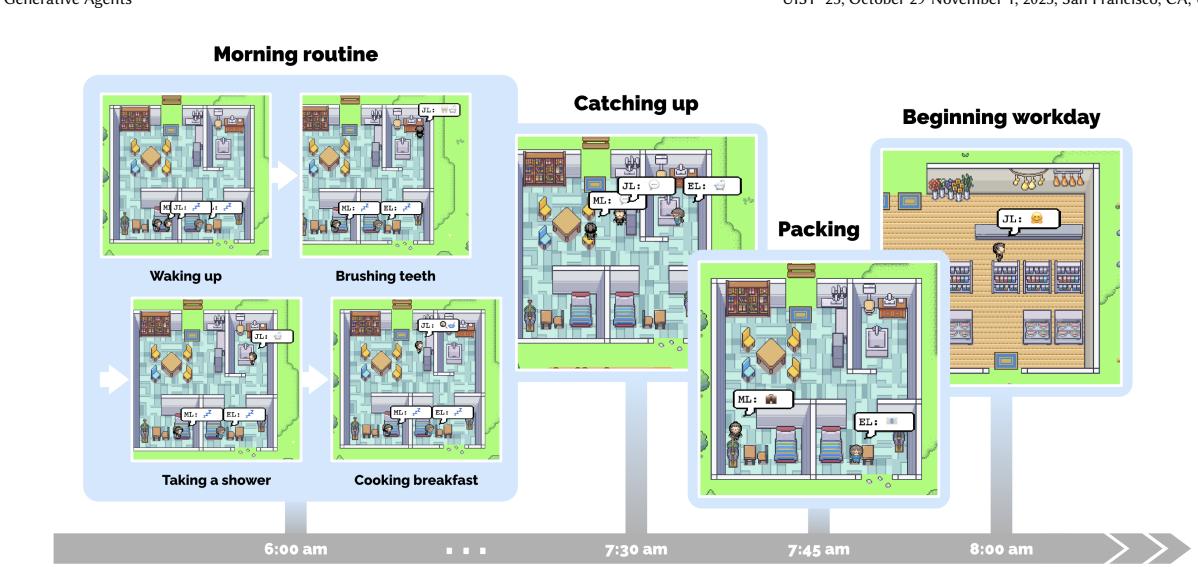
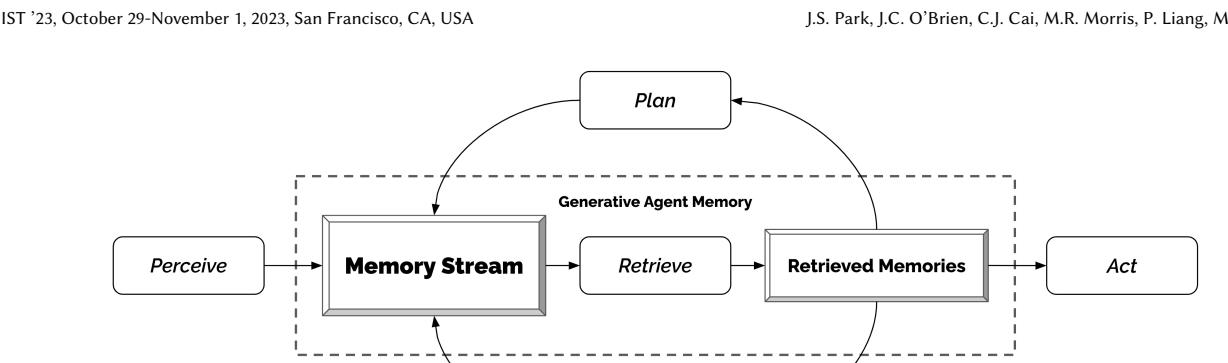


Figure 3: A morning in the life of a generative agent, John Lin. John wakes up around 6 am and completes his morning routine, which includes brushing his teeth, taking a shower, and eating breakfast. He briefly catches up with his wife, Mei, and son, Eddy, before heading out to begin his workday.

Generative Agents



3.1 Agent Avatar and Communication

A community of 25 unique agents inhabits Smallville. Each agent is represented by a simple sprite avatar. We authored one paragraph of natural language description to depict each agent's identity, including their occupation and relationship with other agents, as seen in the sandbox world. The action is displayed on the sandbox interface as a set of emojis, providing an abstract representation of the action from an overhead view. To achieve this, the system utilizes a language model to translate the action into a set of emojis, which appear above each avatar's head in a speech bubble. For example, "Isabella Rodriguez is writing in her journal" is displayed as 📝, while "Isabella Rodriguez is checking her emails" appears as 📧. The complete natural language description of the action can be accessed by clicking on the agent's avatar.

3.1.1 *Inter-Agent Communication.* The agents interact with the world by their actions, and with each other through natural language. At each step of the sandbox engine, the agents output a natural language statement describing their current action, such as "Isabella Rodriguez is writing in her journal", "Isabella Rodriguez is checking her emails", "Isabella Rodriguez is talking with her family on the phone", or "Isabella Rodriguez is getting ready for bed". This statement is then translated into concrete movements that affect the sandbox world. The action is displayed on the sandbox interface as a set of emojis, providing an abstract representation of the action from an overhead view. To achieve this, the system utilizes a language model to translate the action into a set of emojis, which appear above each avatar's head in a speech bubble. For example, "Isabella Rodriguez is writing in her journal" is displayed as 📝, while "Isabella Rodriguez is checking her emails" appears as 📧. The complete natural language description of the action can be accessed by clicking on the agent's avatar.

3.1.2 *Relationship Memory.* Agents in Smallville form new relationships over time and remember their interactions with other agents. For example, at the start, Sam does not know Latoya Williams. While taking a walk in Johnson Park, Sam runs into Latoya, and they introduce themselves. Latoya mentions that she is working on a photography project: "I'm here to take some photos for a project I'm working on." In a later interaction, Sam's interactions with Latoya indicate a memory of that interaction, as he asks "Hi, Latoya. How is your project going?" and she replies "Hi, Sam. It's going well!"

3.1.3 *Coordination.* Generative agents coordinate with each other. Isabella Rodriguez, at Hobbs Cafe, is initialized with an intent to plan a Valentine's Day party from 5 to 7 p.m. on February 14th. From this seed, the agent proceeds to invite friends and customers when she sees them at Hobbs Cafe or elsewhere. Isabella then spends the afternoon of the 13th decorating the cafe for the occasion. Maria, a frequent customer and close friend of Isabella's, arrives at the cafe. Isabella asks for Maria's help in decorating for the party, and Maria agrees. Maria's character description mentions that she has a crush on Klaus. That night, Maria invites Klaus, her secret crush, to join her at the party, and he gladly accepts.

3.1.4 *Memory and Retrieval.* Our generative agent architecture maintains a comprehensive record of the agent's experience. It is a list of memory objects, where each object contains a natural language description, a creation timestamp, and a most recent access timestamp. The most basic element of the memory stream is an observation, which is an event directly perceived by an agent. Common observations include behaviors performed by the agent themselves or behaviors that agents perceive being performed by other agents or non-agent objects. For instance, Isabella Rodriguez, who works at a coffee shop, might perceive the following observations over time: (1) *Isabella* → *Perceives* → *Observation*: "A customer enters the shop." (2) *Isabella* → *Perceives* → *Observation*: "A colleague prepares coffee." (3) *Isabella* → *Perceives* → *Observation*: "A customer orders a latte." (4) *Isabella* → *Perceives* → *Observation*: "A colleague serves the customer." (5) *Isabella* → *Perceives* → *Observation*: "A customer pays for the coffee." (6) *Isabella* → *Perceives* → *Observation*: "A colleague cleans up the counter." (7) *Isabella* → *Perceives* → *Observation*: "A customer leaves the shop." (8) *Isabella* → *Perceives* → *Observation*: "A colleague starts preparing for the next customer." (9) *Isabella* → *Perceives* → *Observation*: "A customer enters the shop." (10) *Isabella* → *Perceives* → *Observation*: "A colleague prepares coffee." (11) *Isabella* → *Perceives* → *Observation*: "A customer orders a latte." (12) *Isabella* → *Perceives* → *Observation*: "A colleague serves the customer." (13) *Isabella* → *Perceives* → *Observation*: "A customer pays for the coffee." (14) *Isabella* → *Perceives* → *Observation*: "A colleague cleans up the counter." (15) *Isabella* → *Perceives* → *Observation*: "A customer leaves the shop." (16) *Isabella* → *Perceives* → *Observation*: "A colleague starts preparing for the next customer."

Generative Agents

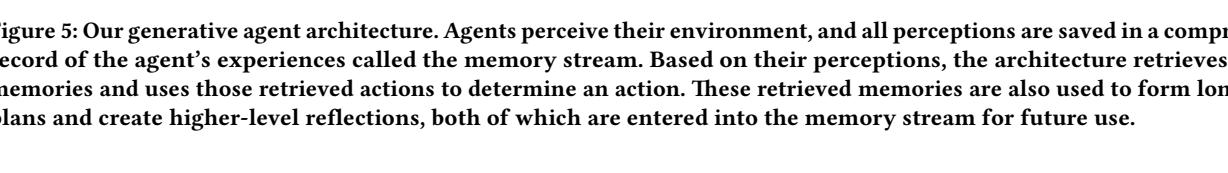


Figure 5: Our generative agent architecture. Agents perceive their environment, and all perceptions are saved in a comprehensive record of the agent's experiences called the memory stream. Based on their perceptions, the architecture retrieves relevant memories and uses those retrieved actions to determine an action. These retrieved memories are also used to form longer-term plans and create higher-level reflections, both of which are entered into the memory stream for future use.

4 GENERATIVE AGENT ARCHITECTURE

Generative agents aim to provide a framework for behavior in an open world: one that can engage in interactions with other agents and react to changes in the environment. Generative agents take larger than what should be described in a prompt, as the full memory stream can distract the model and does not even currently fit into the limited context window. Consider the Isabella agent answering the question, "What are you passionate about these days? Summarizing all of Isabella's experiences to fit in the limited context window of the language model produces an uninformative response, where Isabella discusses topics such as collaborations for events and projects and cleanliness and organization in a cafe. Instead of summarizing, the memory stream described below surface relevant memories, resulting in a more informative and specific response that mentions Isabella's passion for making people feel welcome and included, planning events and creating an atmosphere that people can enjoy, such as the Valentine's Day party.

Approach: The memory stream maintains a comprehensive record of the agent's experience. It is a list of memory objects, where each object contains a natural language description, a creation timestamp, and a most recent access timestamp. The most basic element of the memory stream is an observation, which is an event directly perceived by an agent. Common observations include behaviors performed by the agent themselves or behaviors that agents perceive being performed by other agents or non-agent objects. For instance, Isabella Rodriguez, who works at a coffee shop, might perceive the following observations over time: (1) *Isabella* → *Perceives* → *Observation*: "A customer enters the shop." (2) *Isabella* → *Perceives* → *Observation*: "A colleague prepares coffee." (3) *Isabella* → *Perceives* → *Observation*: "A customer orders a latte." (4) *Isabella* → *Perceives* → *Observation*: "A colleague serves the customer." (5) *Isabella* → *Perceives* → *Observation*: "A customer pays for the coffee." (6) *Isabella* → *Perceives* → *Observation*: "A colleague cleans up the counter." (7) *Isabella* → *Perceives* → *Observation*: "A customer leaves the shop." (8) *Isabella* → *Perceives* → *Observation*: "A colleague starts preparing for the next customer." (9) *Isabella* → *Perceives* → *Observation*: "A customer enters the shop." (10) *Isabella* → *Perceives* → *Observation*: "A colleague prepares coffee." (11) *Isabella* → *Perceives* → *Observation*: "A customer orders a latte." (12) *Isabella* → *Perceives* → *Observation*: "A colleague serves the customer." (13) *Isabella* → *Perceives* → *Observation*: "A customer pays for the coffee." (14) *Isabella* → *Perceives* → *Observation*: "A colleague cleans up the counter." (15) *Isabella* → *Perceives* → *Observation*: "A customer leaves the shop." (16) *Isabella* → *Perceives* → *Observation*: "A colleague starts preparing for the next customer."

Communicating Research

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