

Research Skills

Figures & Visuals

Andrea Bajcsy

abajcsy@cmu.edu

Join our Discord to test V3!

Craiyon v2

Free online AI image generator from text



an academic figure for a computer vision research paper



Draw



an academic **graph** for a computer vision paper

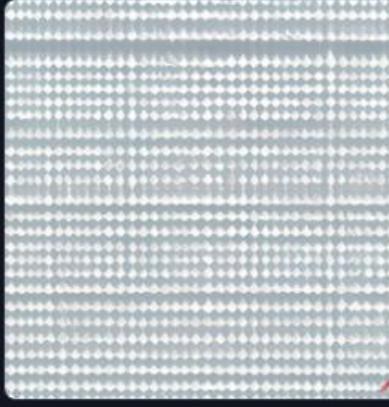
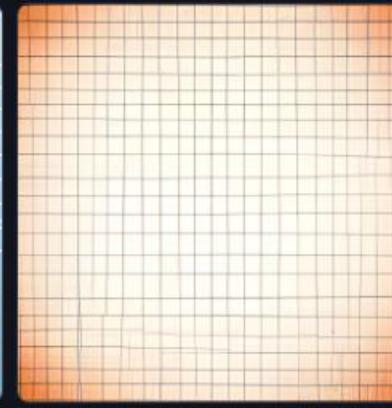
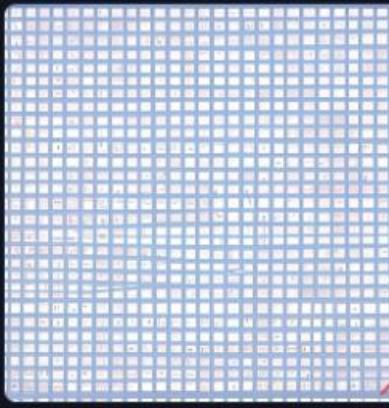
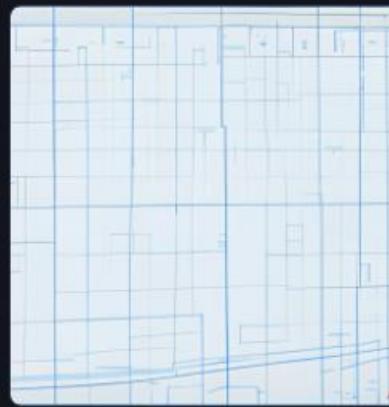
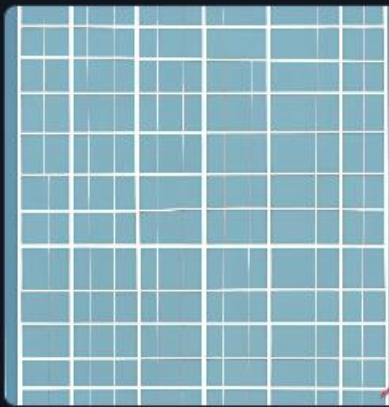


Draw

an academic graph for a computer vision paper



Draw



scatterplot of sine wave

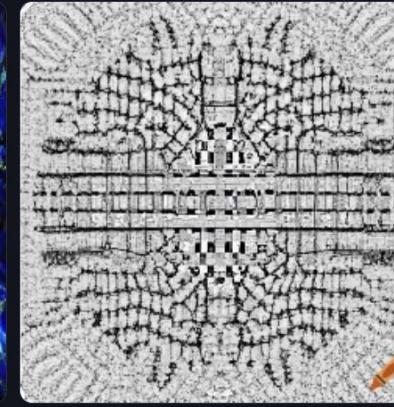
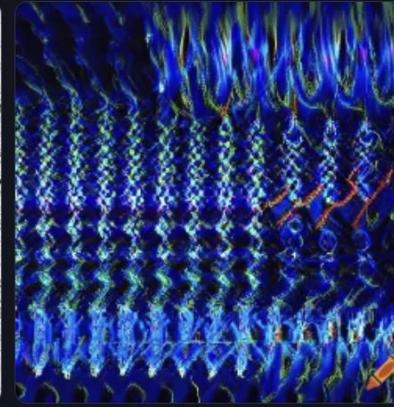
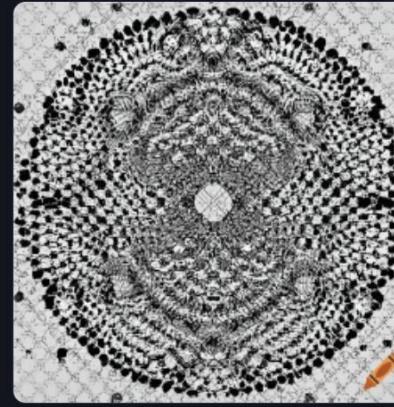
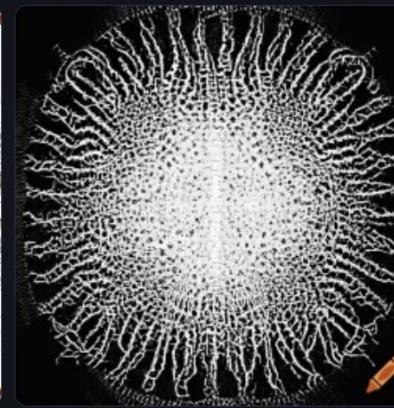
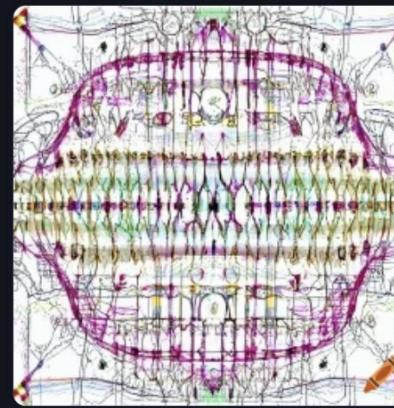
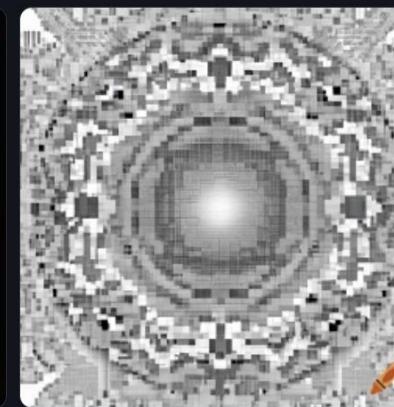
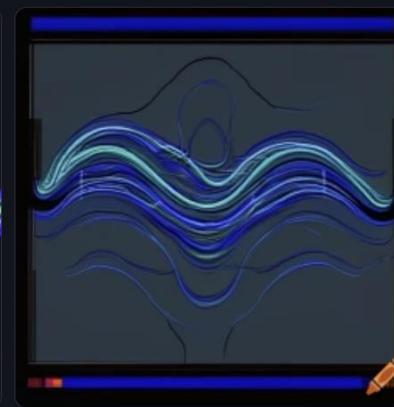
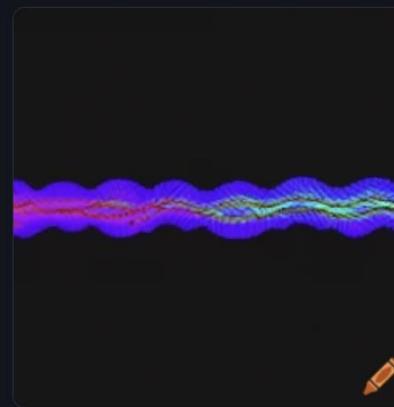


Draw

scatterplot of sine wave

(

 Draw



Default (GPT-3.5)

AB

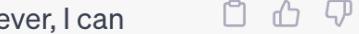
draw a plot of a sine wave

AB

draw a plot of a sine wave



I'm a text-based AI and unable to create visual drawings or plots directly. However, I can describe how to create a simple sine wave plot using Python and the popular matplotlib library. You can run this code in a Python environment to generate a sine wave plot:

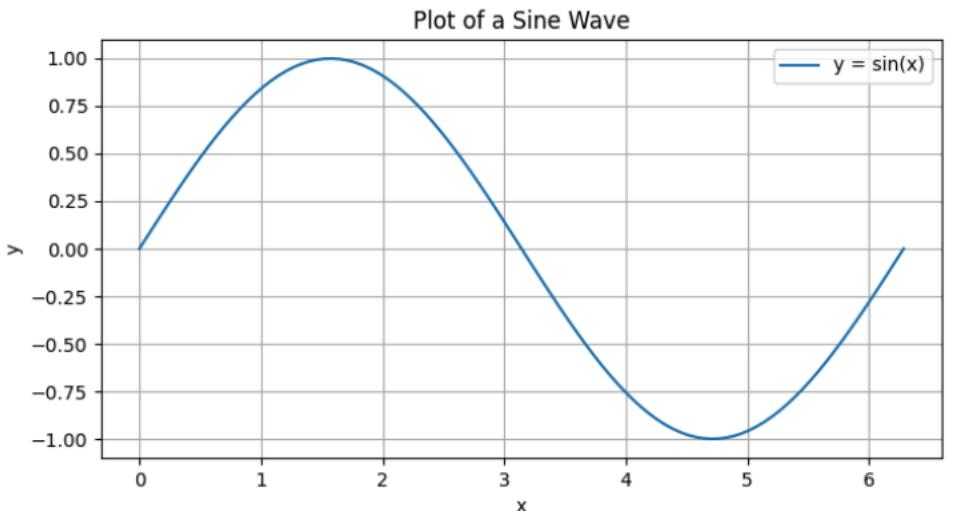


python

Copy code

```
import numpy as np
import matplotlib.pyplot as plt

# Create a time array
```



```
* np.pi, 1000) # Generate values from 0 to 2*pi

u can adjust the amplitude
u can adjust the frequency
* np.sin(frequency * t)

, 4))
, label='Sine Wave')
```



draw a figure that could be at the top of a computer vision research paper



draw a figure that could be at the top of a computer vision research paper



Certainly! For a computer vision research paper, the top figure often highlights the core concept or architecture of the proposed method. Let's create a simple conceptual diagram of an object detection pipeline. This could show how an image is processed through a convolutional neural network (CNN) to produce bounding boxes and class labels.

Here's a basic example of such a diagram:

```
python Copy code
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle

# Create a new figure
fig, ax = plt.subplots(figsize=(10, 4))

# Draw the image block
ax.add_patch(Rectangle((0.1, 0.3), 0.15, 0.4, edgecolor='black', facecolor='white'))
plt.text(0.175, 0.5, 'Input Image', ha='center', va='center', fontsize=12)

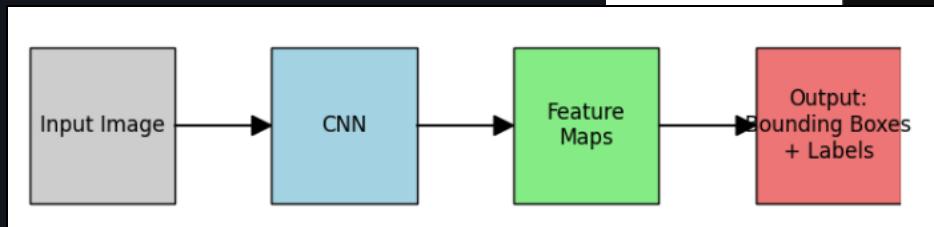
# Draw the CNN block
ax.add_patch(Rectangle((0.35, 0.3), 0.15, 0.4, edgecolor='black', facecolor='lightblue'))
plt.text(0.425, 0.5, 'CNN', ha='center', va='center', fontsize=12)

# Draw the output block
ax.add_patch(Rectangle((0.6, 0.3), 0.15, 0.4, edgecolor='black', facecolor='pink'))
plt.text(0.675, 0.5, 'Output: Bounding Boxes + Labels', ha='center', va='center', fontsize=12)

# Arrows
arrow1 = plt.Arrow(0.25, 0.5, 0.35, 0.5, head_width=0.05, head_length=0.02, fc='black')
arrow2 = plt.Arrow(0.55, 0.5, 0.65, 0.5, head_width=0.05, head_length=0.02, fc='black')

# Labels
label1 = plt.Text(0.4, 0.7, 'arrows from Input Image to CNN')
label2 = plt.Text(0.5, 0.7, 'the Feature Maps block')

# ChatGPT watermark
chatgpt_watermark = plt.Text(0.5, 0.9, 'ChatGPT')
```



Instead of hoping to auto-generate our academic figures...
... let's talk about the *process of designing* them

Claim:

The *process of graphic design* is a
multi-objective optimal control problem

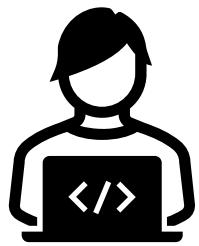
i.e. sequential decision-making problem!

The *process of graphic design* is a
multi-objective optimal control problem

Thinking of it this way:

1. **Reduction to a known problem:** Makes the daunting challenge of design more “familiar”
2. **Reward design:** Lets us think about what **objectives** we seek to optimize with our designs
3. **Approximations:** Lets us derive practical design heuristics

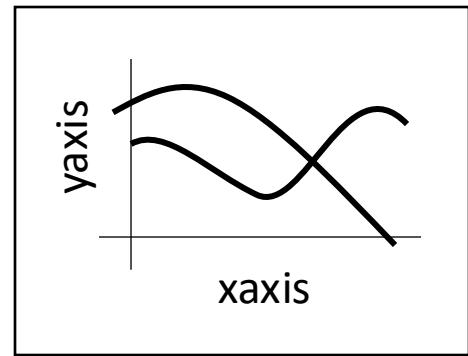
Formalizing the Graphic Design Process



*You
(the designer)*

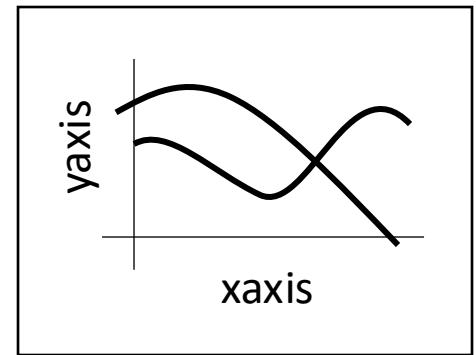
state χ

the design you see



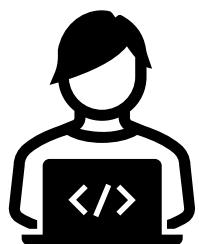
state x

the design you see

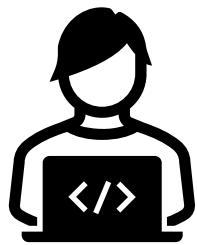


control u

$\in \{change_{font}, change_{color},$
 $add_{line}, add_{text}, \dots\}$



You
(the designer)



You
(the designer)

state

x

the design you see

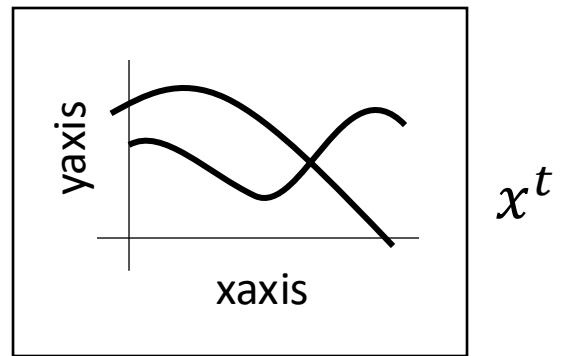
control

u

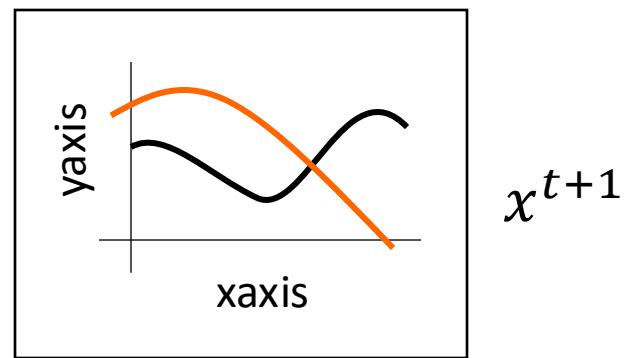
$\in \{change_{font}, change_{color},$
 $add_{line}, add_{text}, \dots\}$

dynamics

$$x^{t+1} = f(x^t, u^t)$$



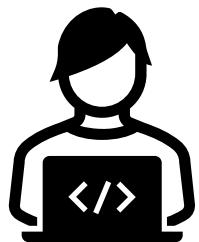
$$u^t = change_{color}$$



$$\max_{u^{0:T}}$$



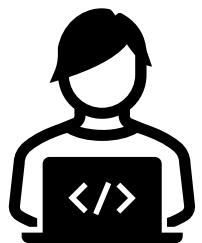
*sequence of design
decisions*



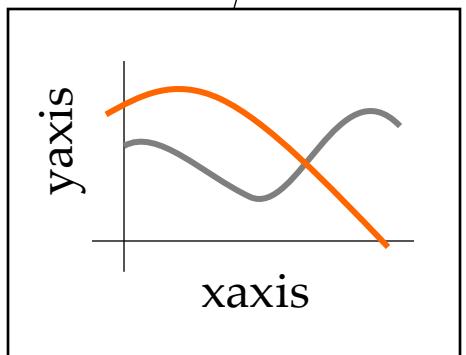
*You
(the designer)*

*Information density
of final design*

$$\max_{u^{0:T}} \text{Info}(x^T)$$

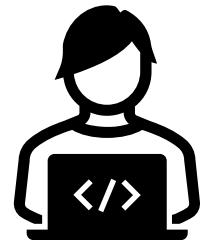


You
(the designer)



*Information density
of final design*

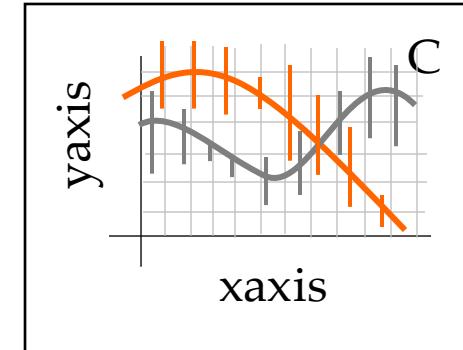
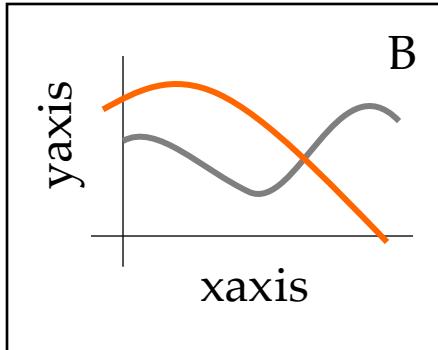
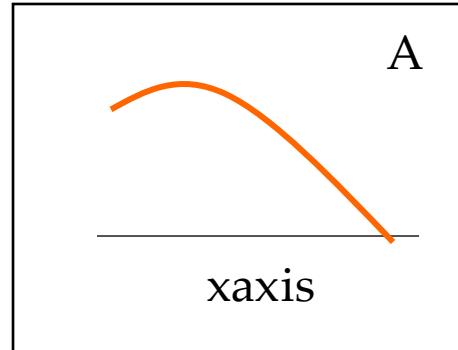
$$\max_{u^{0:T}} \text{Info}(x^T)$$



You
(the designer)



Engineers typically don't
have trouble with this ☺



$$\text{Info}(x_A^T)$$

<

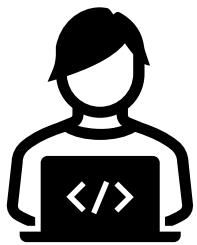
$$\text{Info}(x_B^T)$$

<

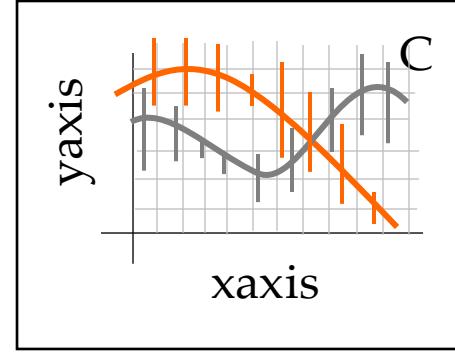
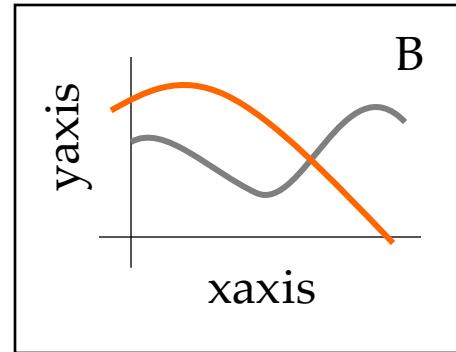
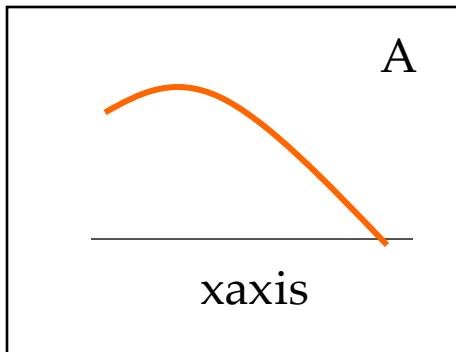
$$\text{Info}(x_C^T)$$

*Viewer's cognitive load when
parsing the final design*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



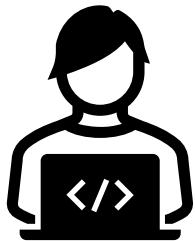
*You
(the designer)*



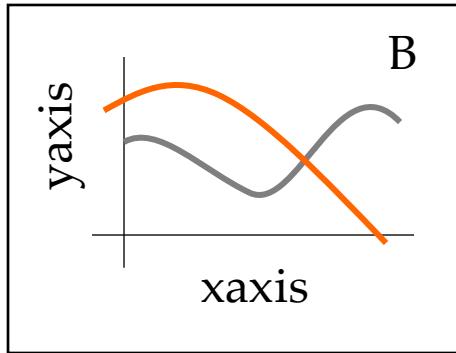
The viewer

*Viewer's cognitive load when
parsing the final design*

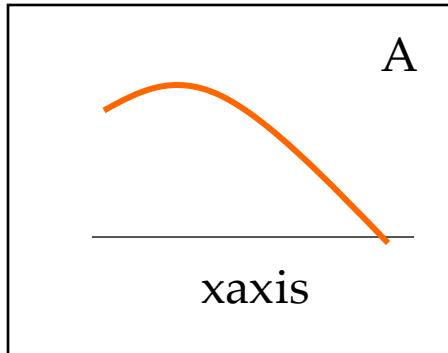
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



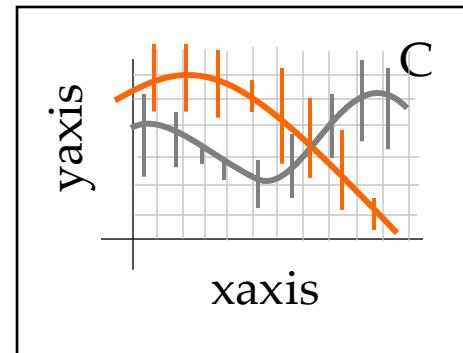
You
(the designer)



$\text{ViewerEffort}(x_B^T)$



$\text{ViewerEffort}(x_A^T)$



$\text{ViewerEffort}(x_C^T)$

Hard to tell what y-axis is

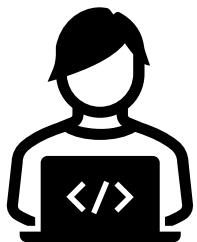
Hard to focus on trends



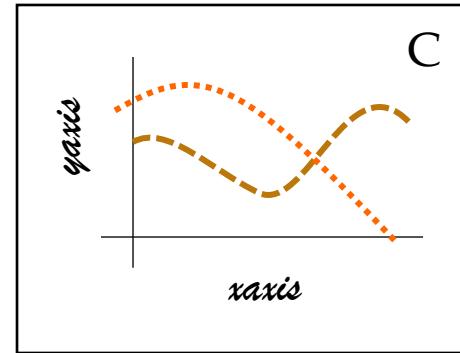
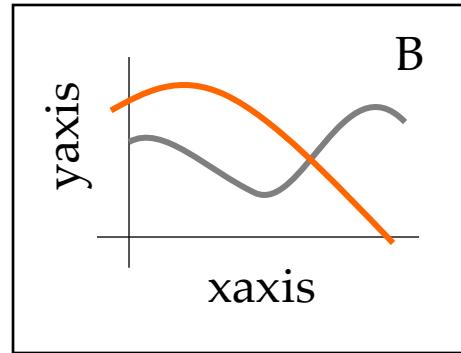
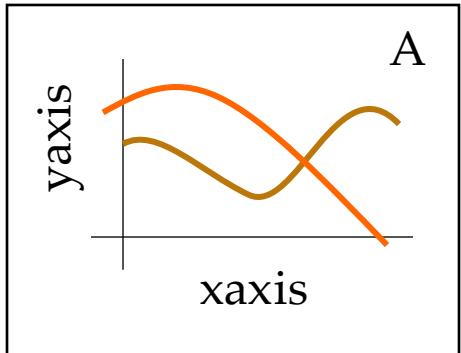
The viewer

*Viewer's cognitive load when
parsing the final design*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



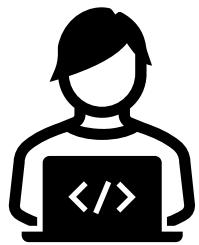
*You
(the designer)*



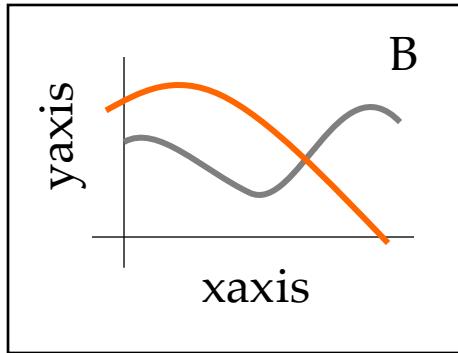
The viewer

*Viewer's cognitive load when
parsing the final design*

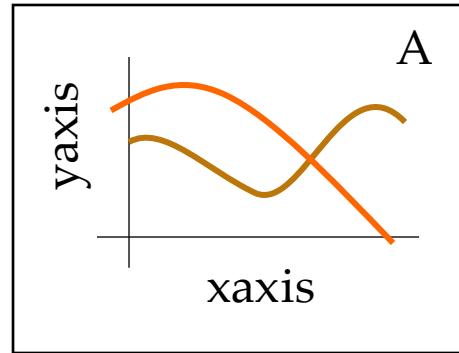
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



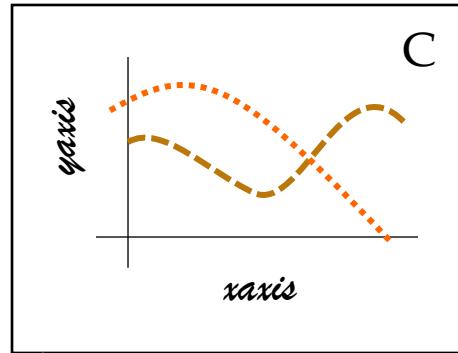
You
(the designer)



$\text{ViewerEffort}(x_B^T)$



$\text{ViewerEffort}(x_A^T)$



$\text{ViewerEffort}(x_C^T)$

Line colors blend together

*Hard to read font, line
color and pattern*



The viewer

Figures from academic papers...

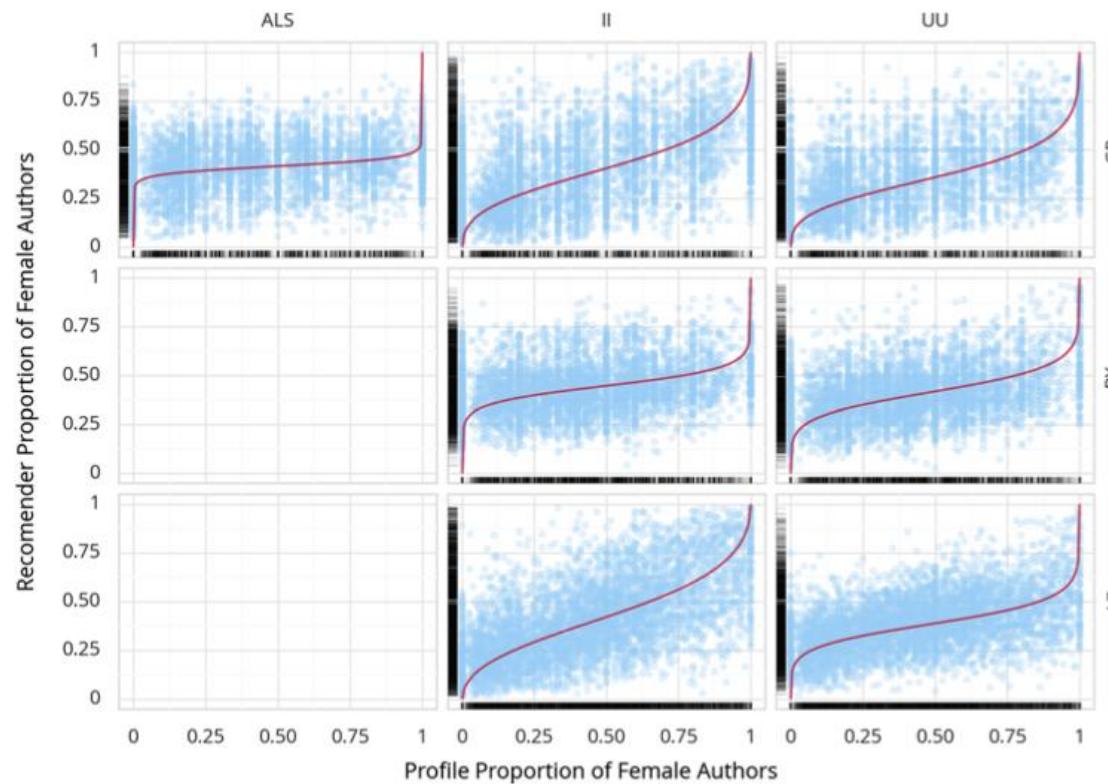


Fig. 12 Scatter plots and regression curves for explicit feedback recommender response to individual users. Points are observed y/n proportions; curves are regression lines transformed from log-odds to proportions. Rug plots show marginal distributions

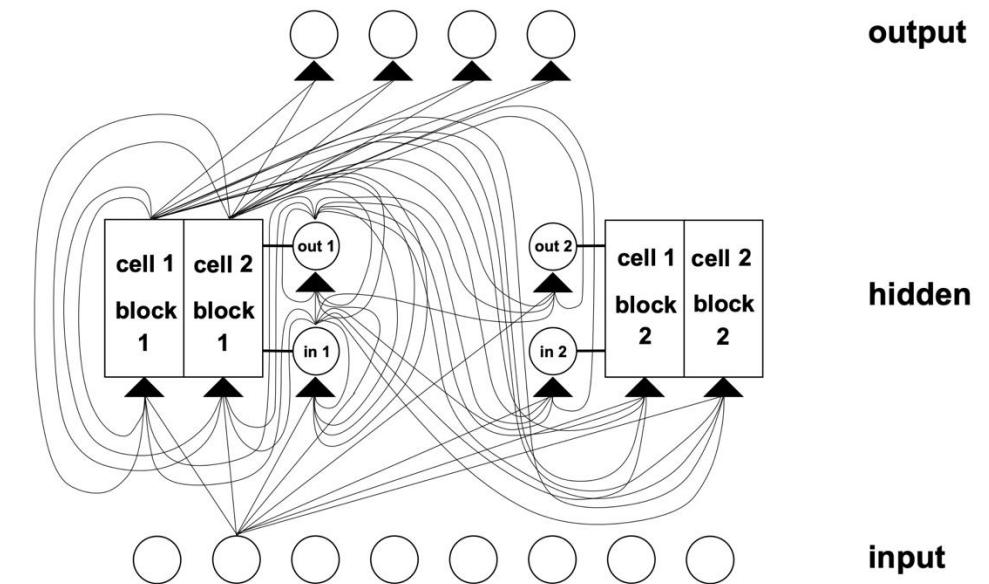


Figure 2: Example of a net with 8 input units, 4 output units, and 2 memory cell blocks of size 2. *in1* marks the input gate, *out1* marks the output gate, and *cell1/block1* marks the first memory cell of block 1. *cell1/block1*'s architecture is identical to the one in Figure 1, with gate units *in1* and *out1* (note that by rotating Figure 1 by 90 degrees anti-clockwise, it will match with the corresponding parts of Figure 1). The example assumes dense connectivity: each gate unit and each memory cell see all non-output units. For simplicity, however, outgoing weights of only one type of unit are shown for each layer. With the efficient, truncated update rule, error flows only through connections to output units, and through fixed self-connections within cell blocks (not shown here — see Figure 1). Error flow is truncated once it “wants” to leave memory cells or gate units. Therefore, no connection shown above serves to propagate error back to [↑] which the connection originates (except for connections to output units), although

$\text{Info}(x^T)$ – $\text{ViewerEffort}(x^T)$



The viewer

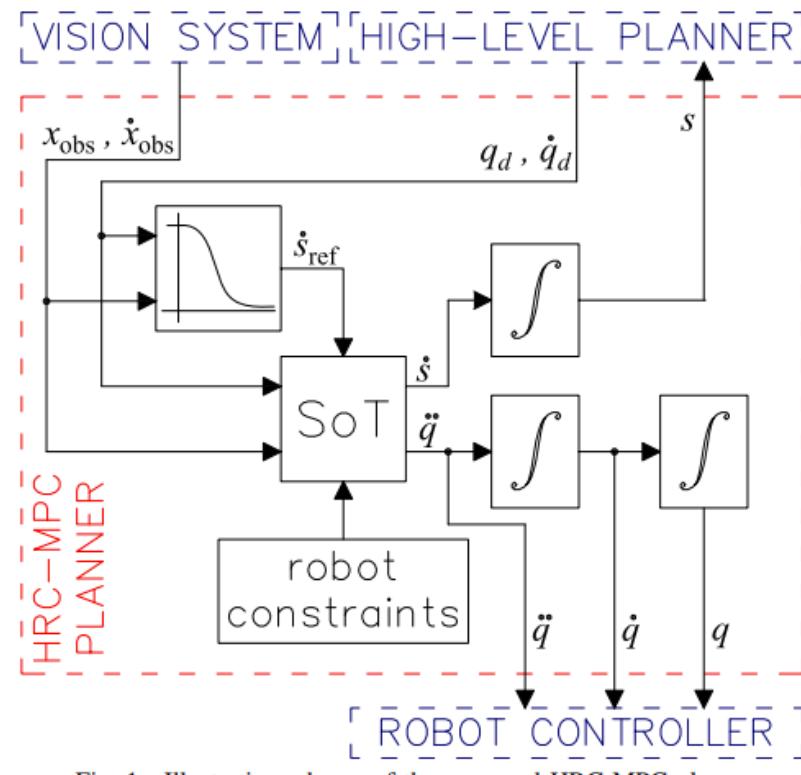
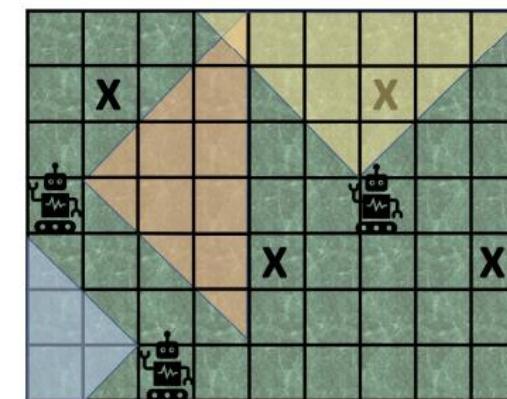
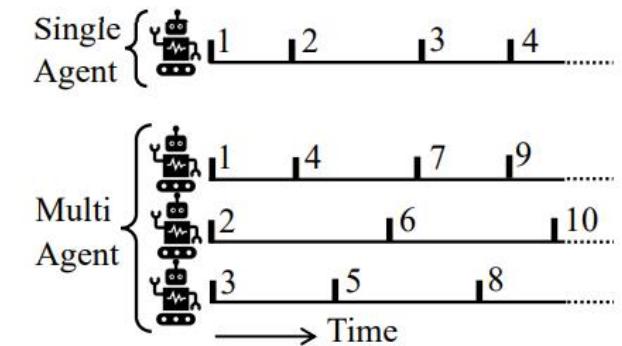


Fig. 1. Illustrative scheme of the proposed HRC-MPC planner.



(a) Multi-Agent Active search



(b) Single vs. Multi



The viewer

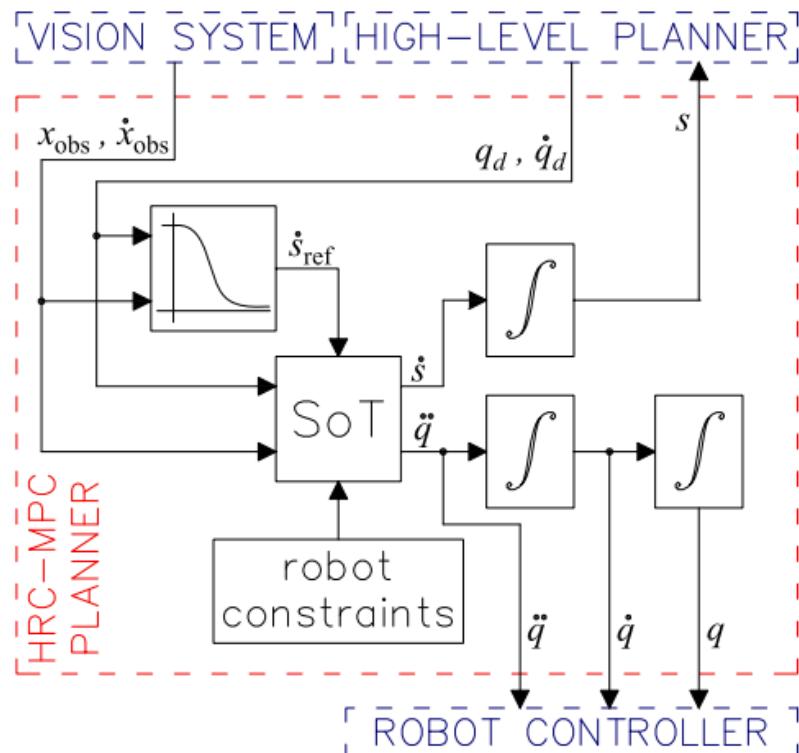
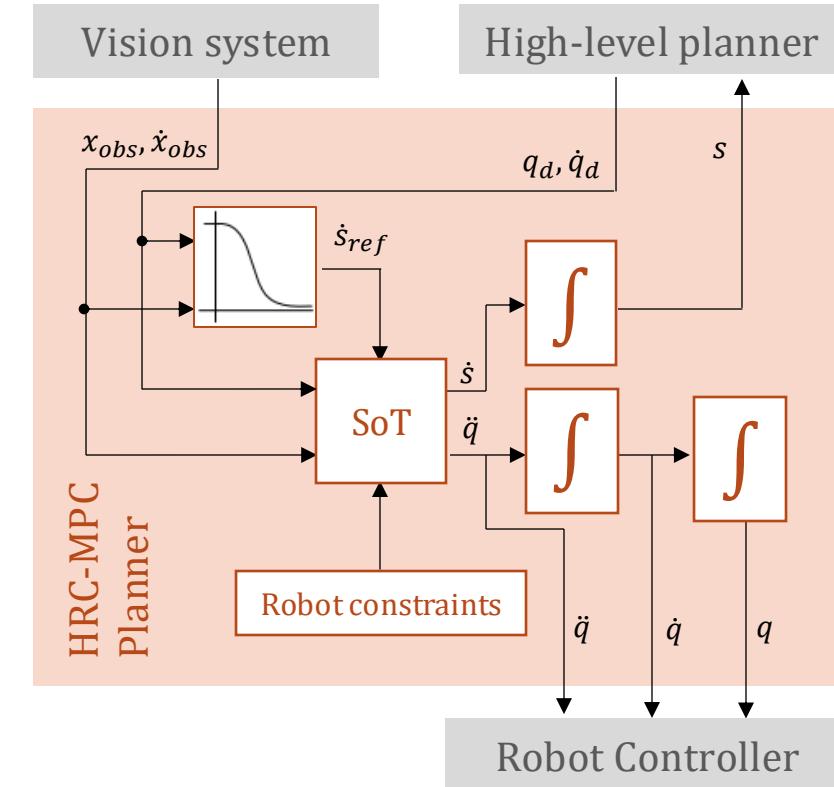


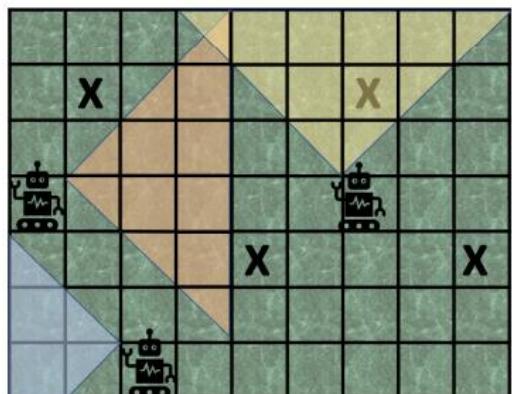
Fig. 1. Illustrative scheme of the proposed HRC-MPC planner.



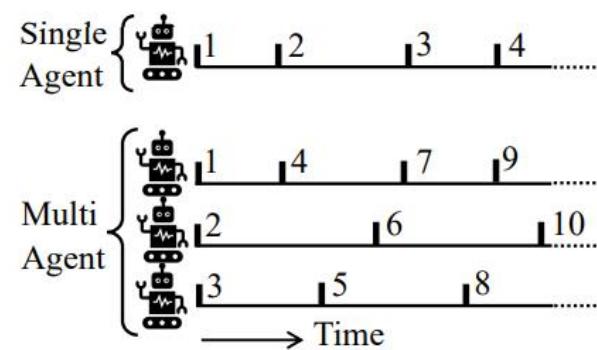
Easier to read font

Color highlights the main contribution

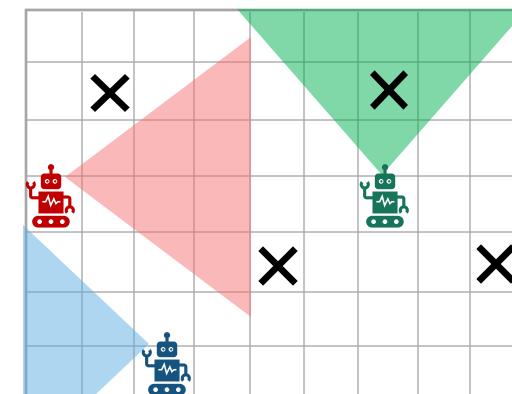
Color blocking shows relationships



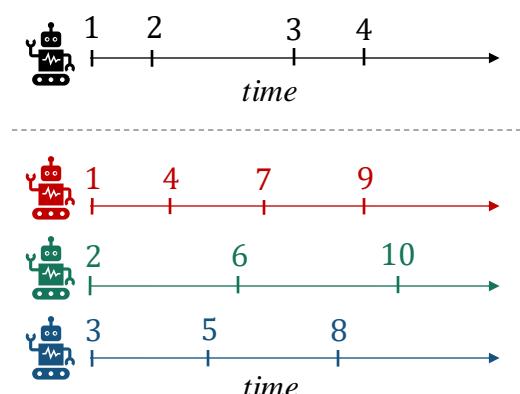
(a) Multi-Agent Active search



(b) Single vs. Multi



(a) Multi-Agent Active search



(b) Single vs. Multi

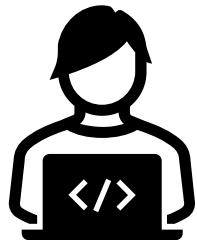
No unnecessary background

Color has meaning (agent identity)

Remove unnecessary words

How do we design this reward function?

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

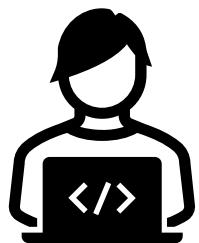


You
(the designer)



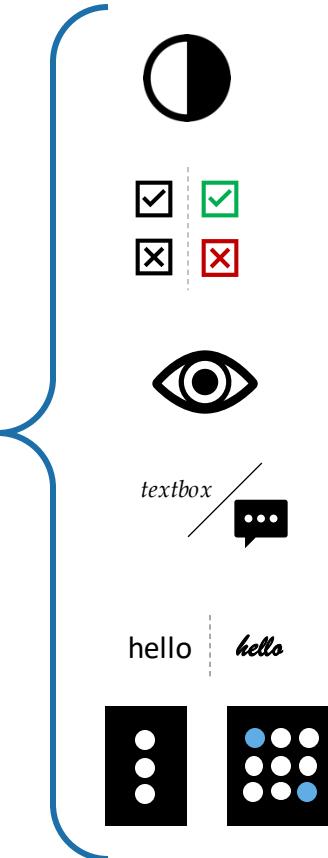
The viewer

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



You
(the designer)

low ViewerEffort



contrasting colors

meaningful colors

reduce “visual clutter”

strategic text, more icons

readable fonts

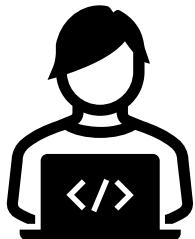
alignment, balance-symmetry



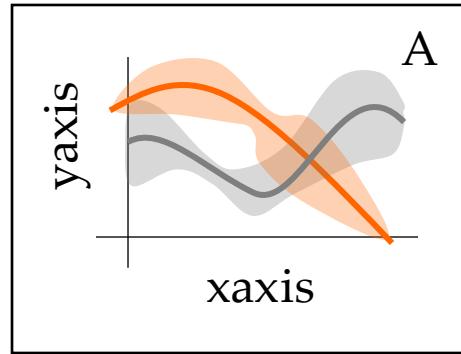
The viewer

Relationship between Info and ViewerEffort

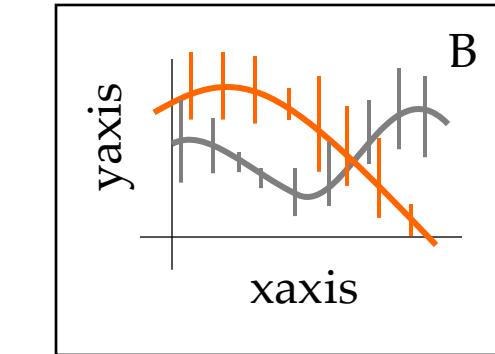
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



*You
(the designer)*



$$\text{Info}(x_A^T) = \text{Info}(x_B^T)$$
$$\text{ViewerEffort}(x_A^T) \wedge \text{ViewerEffort}(x_B^T)$$



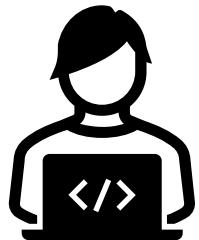
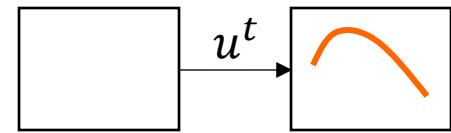
The viewer

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

s. t.

$$x^{t+1} = f(x^t, u^t)$$

dynamics



You
(the designer)



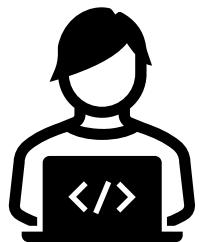
The viewer

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

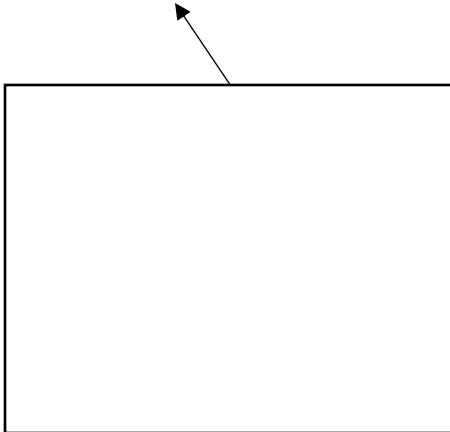
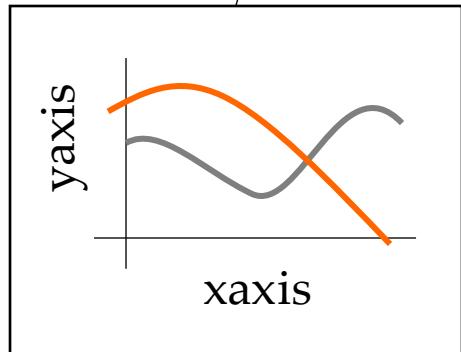
s.t.

$$x^{t+1} = f(x^t, u^t)$$

$$x^0 = x_{blank}$$



You
(the designer)



The viewer

*This is a sequential decision-making problem
(i.e., optimal control problem)!*



$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\begin{aligned} s.t. \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{\text{blank}} \end{aligned}$$

But, challenging because design space is huge...

Tractable Approximations

Dimensionality Reduction

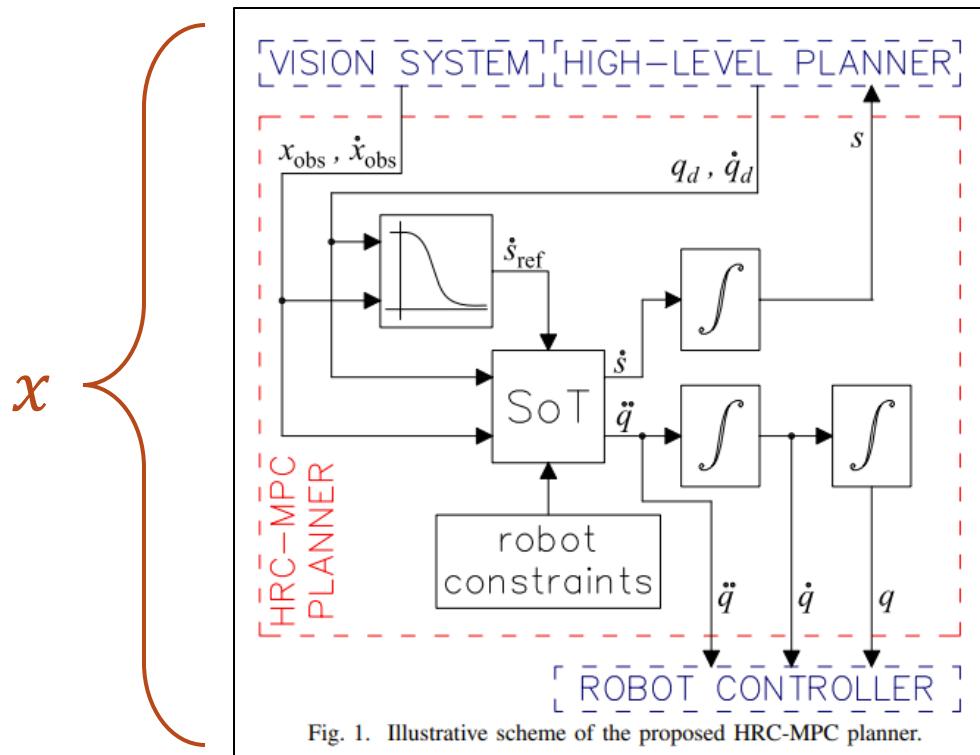
Reduce the size of the design “state” space

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\begin{aligned} s.t. \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{blank} \end{aligned}$$

Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

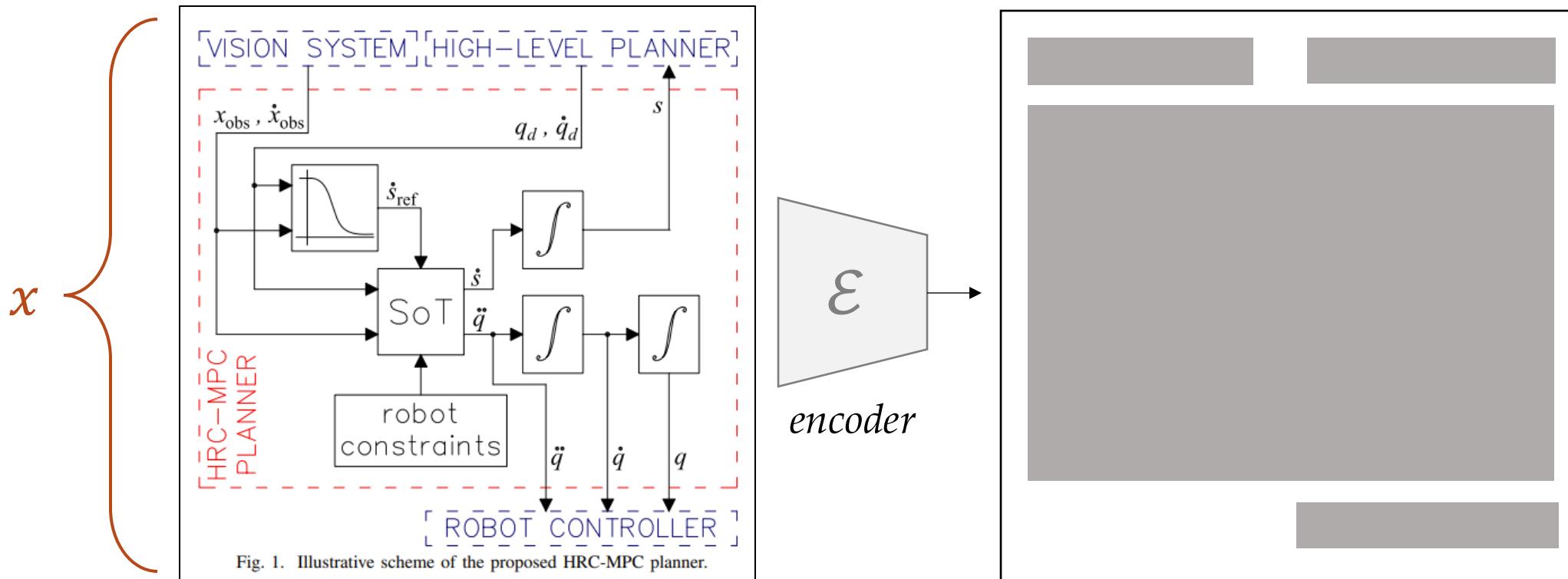


Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$

encoded design *encoded design*

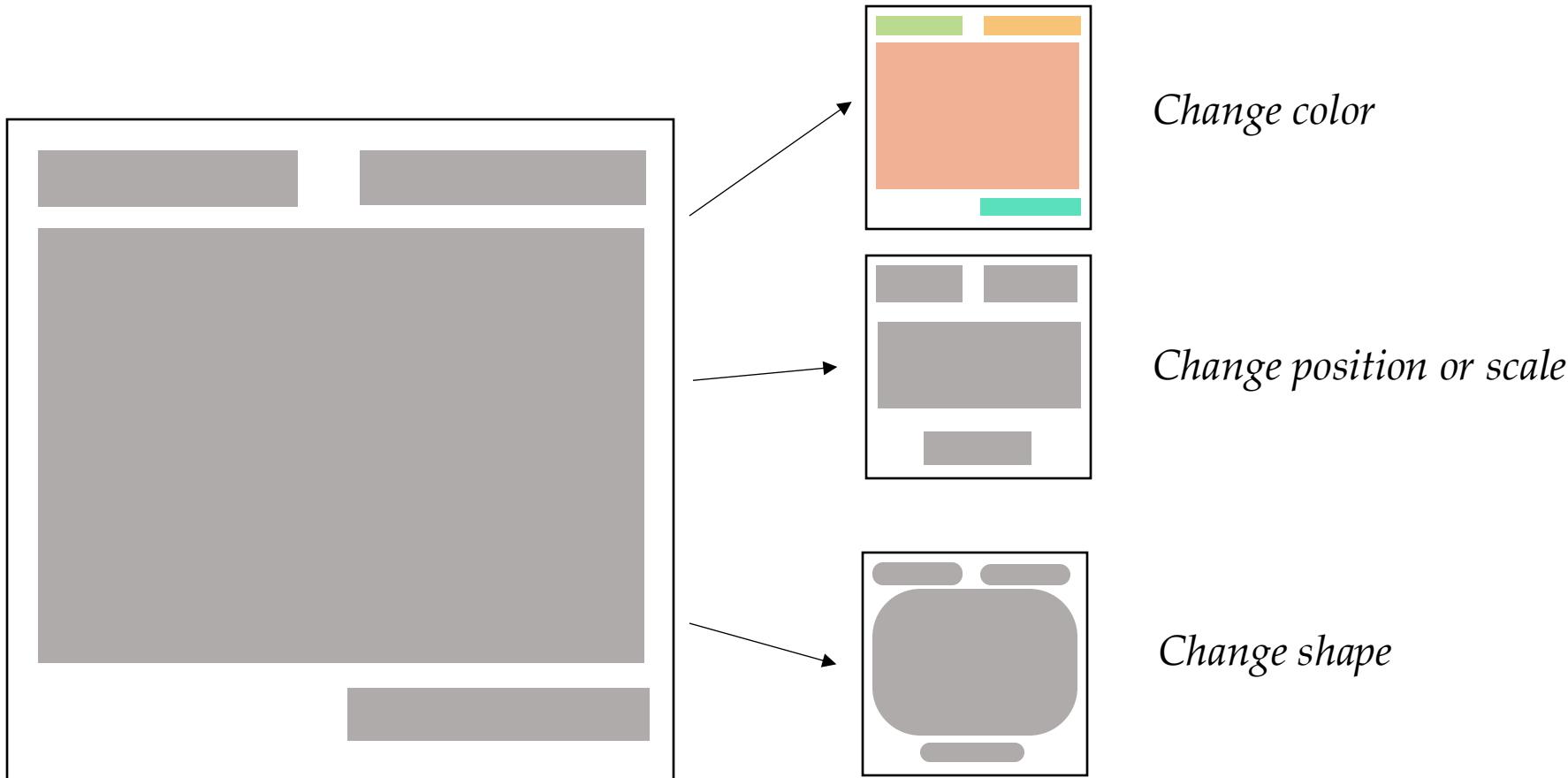
$\mathcal{E}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ where $n \gg m$



Consider only the “primitives” of the design

Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$



It's easier to reason about how “actions” influence the design

Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$

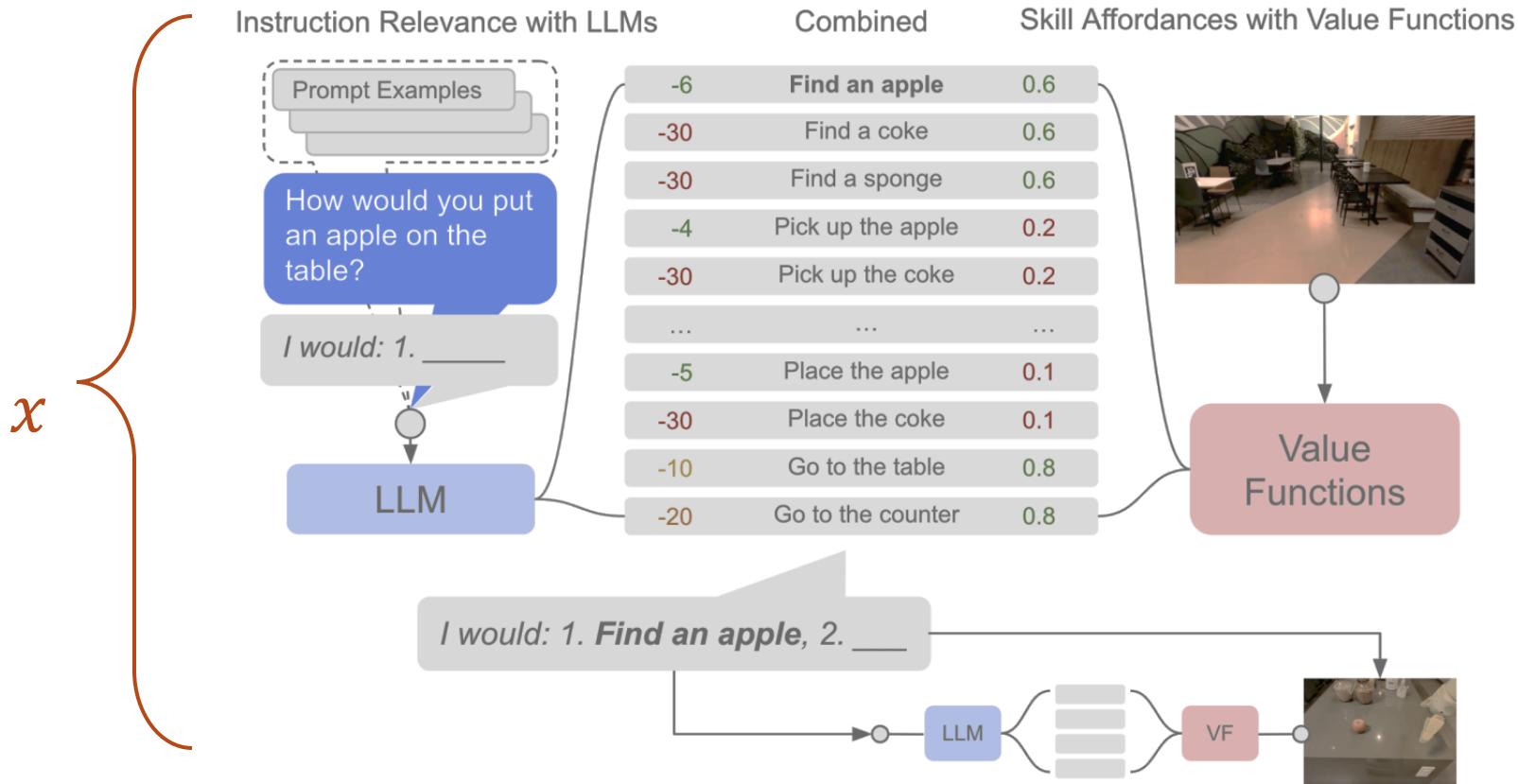
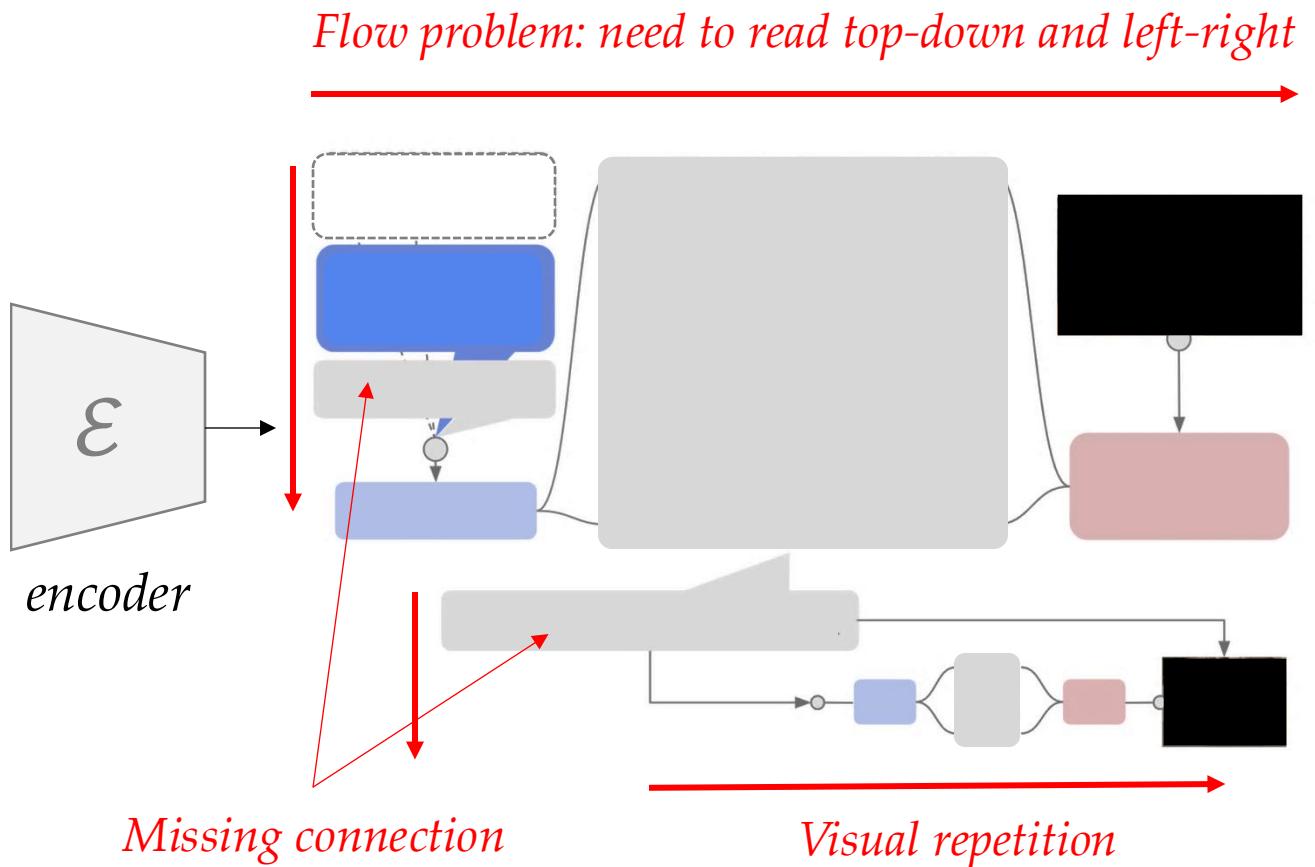
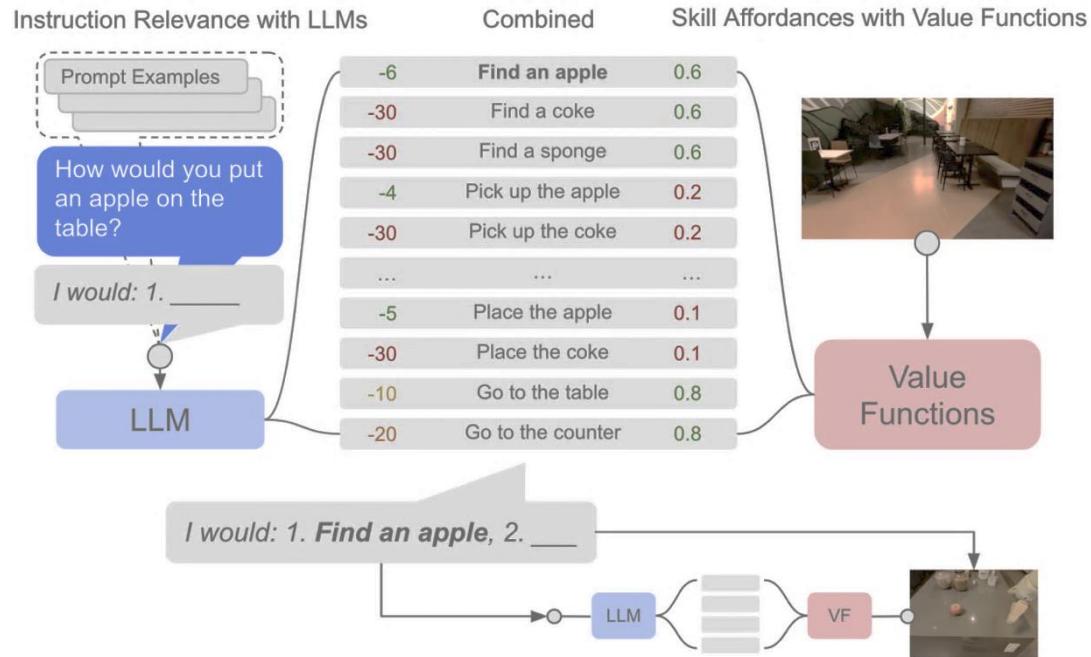


Figure 3: Given a high-level instruction, SayCan combines probabilities from a LLM (the probability that a skill is useful for the instruction) with the probabilities from a value function (the probability of successfully executing said skill) to select the skill to perform. This emits a skill that is both possible and useful. The process is repeated by appending the skill to the response and querying the models again, until the output step is to terminate. Appendix Figures 12 and 2 focus on the LLM and VFS components.

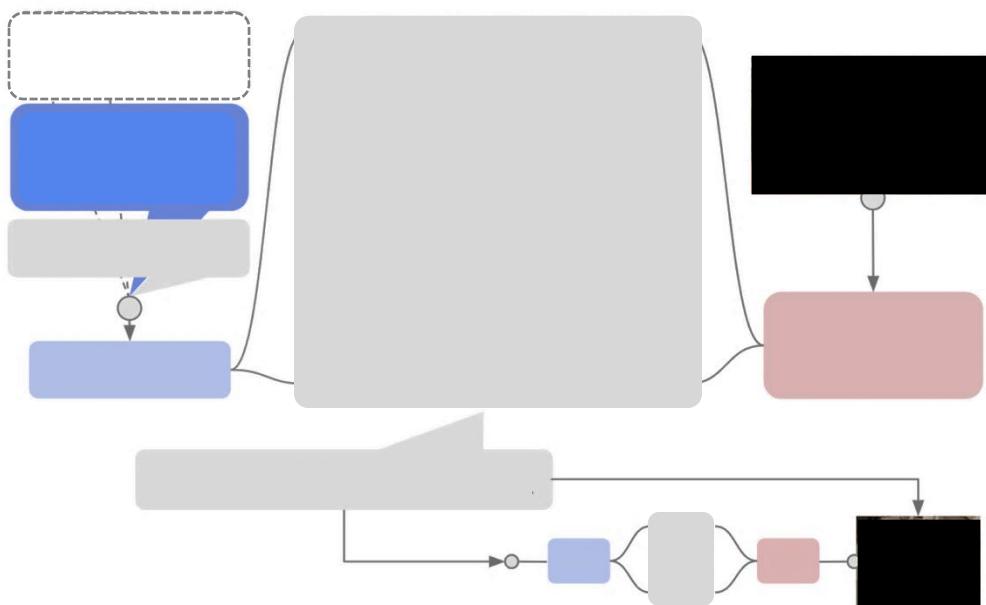
Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$



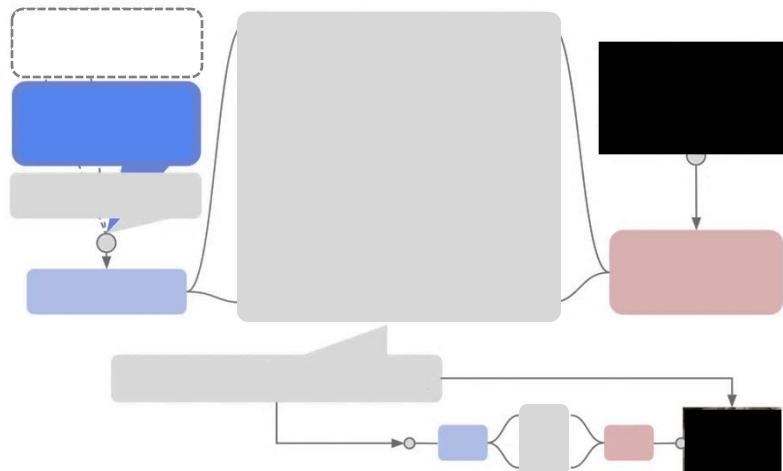
Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$

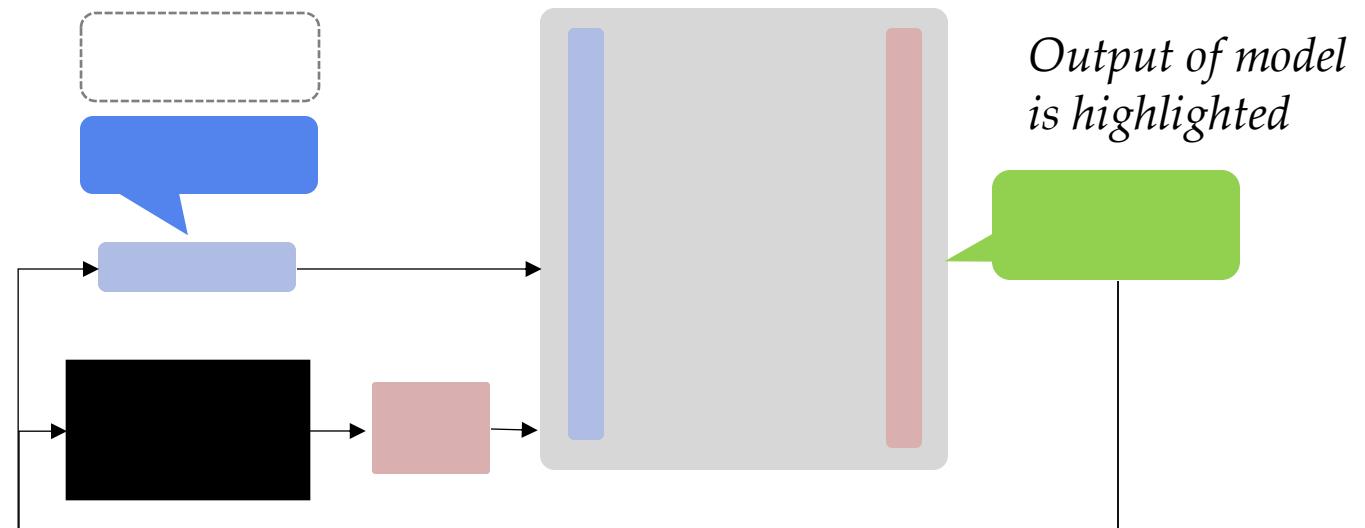


Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$



Use color to connect inner and outer components



Use “feedback” arrows to show the repetition instead of repeating the graphic

Minimize Control Effort

...by sketching before generating final figures

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

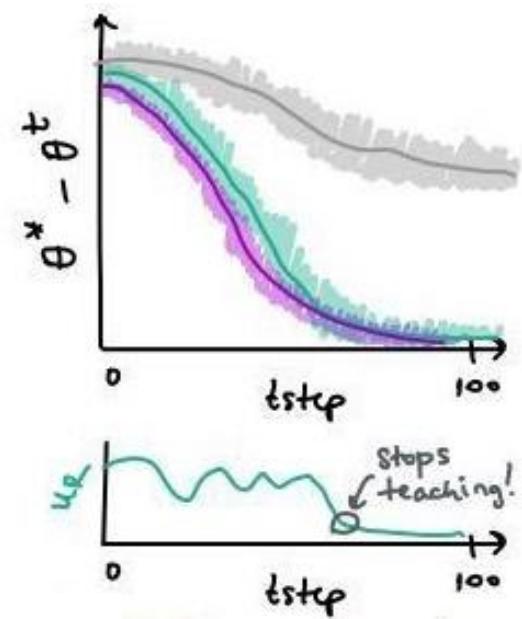
$$s.t. \quad x^{t+1} = f(x^t, u^t)$$

$$x^0 = x_{blank}$$

Minimize Control Effort

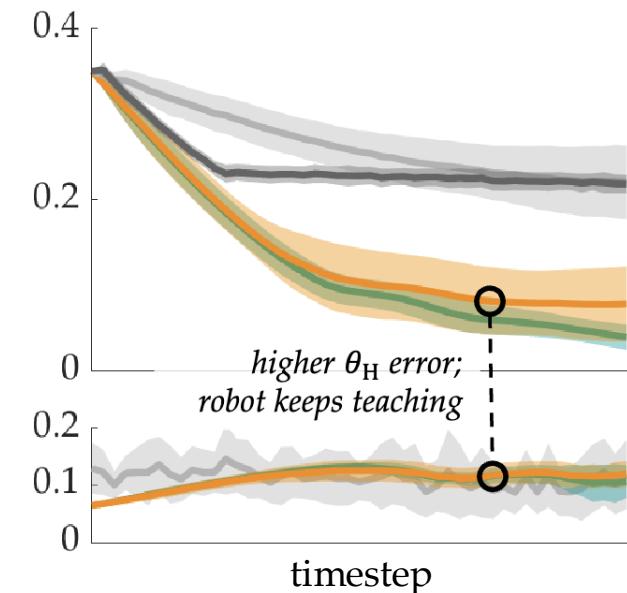
...by sketching before programmatically generating figures

Sketch



— gt model — learned model — no robot

Final Design

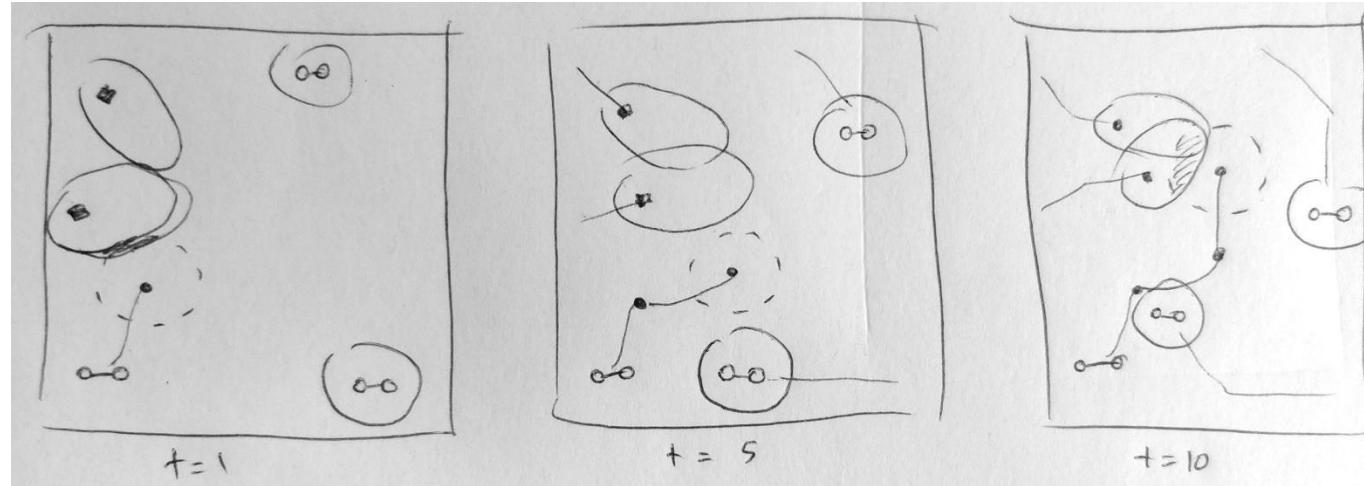


— active teach (ours) — oracle — passive learn — random

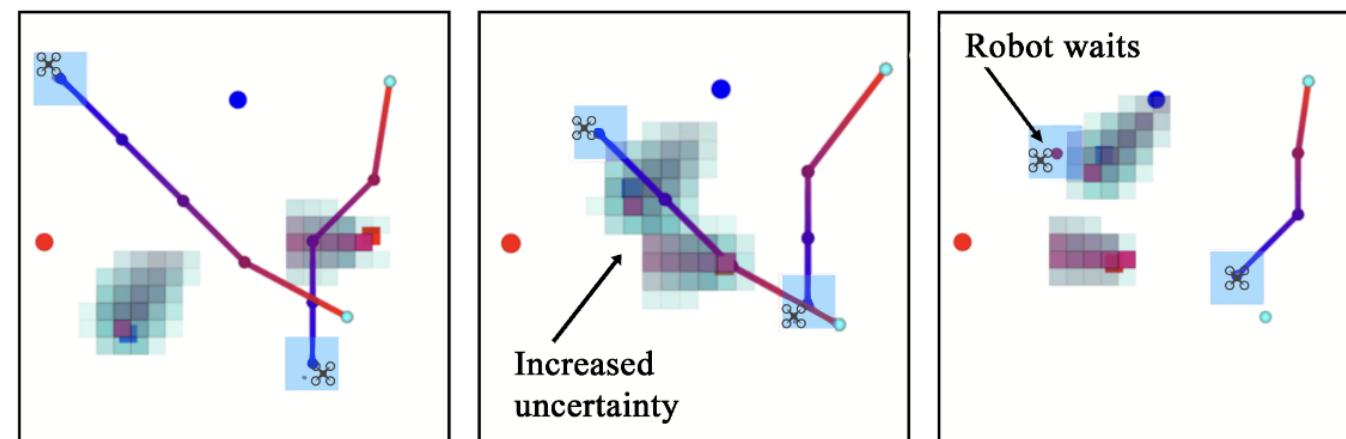
Minimize Control Effort

...by sketching before running expensive experiments

Sketch



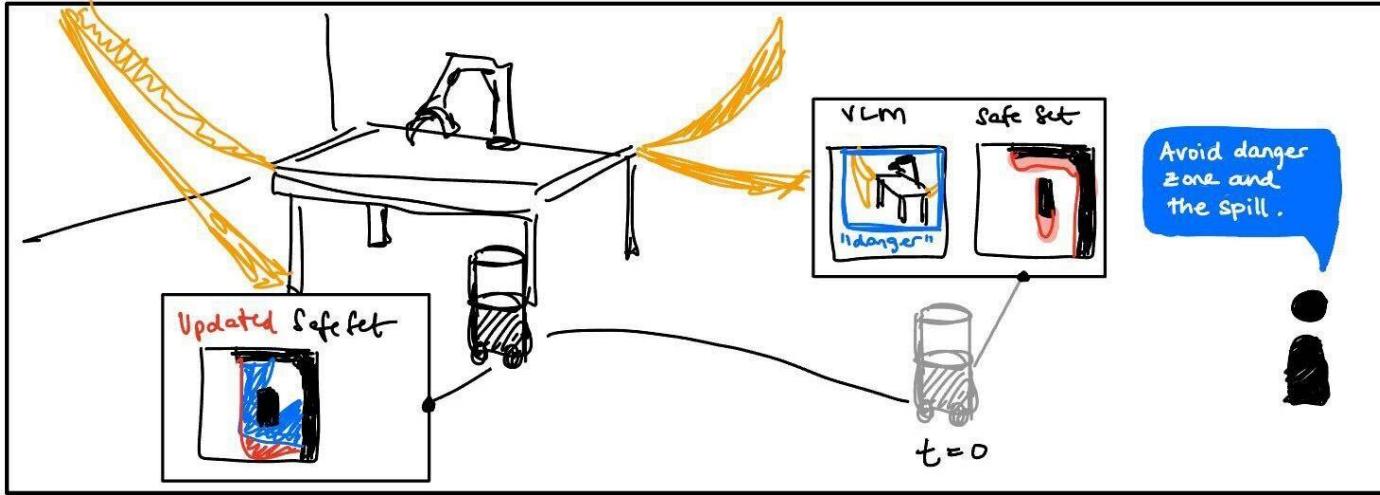
Final Design



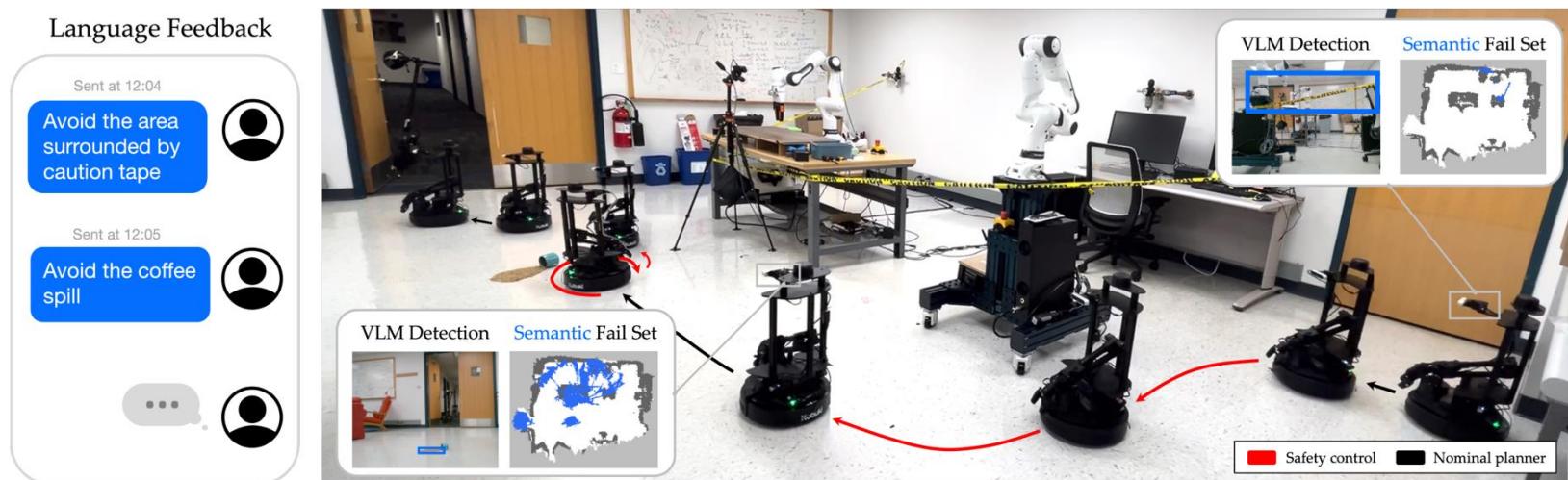
Minimize Control Effort

...by sketching before running expensive experiments

Sketch



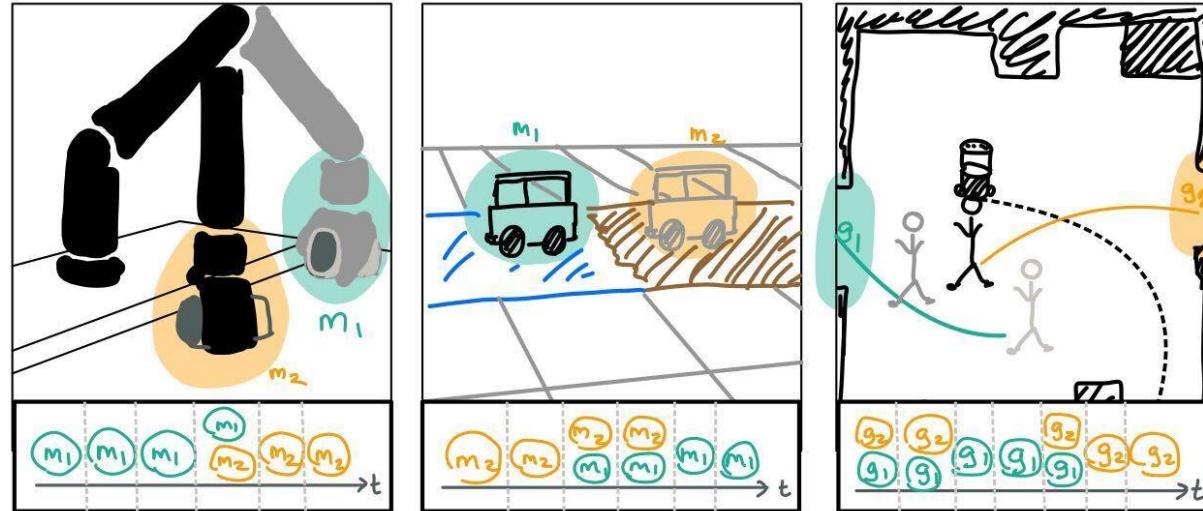
Final Design



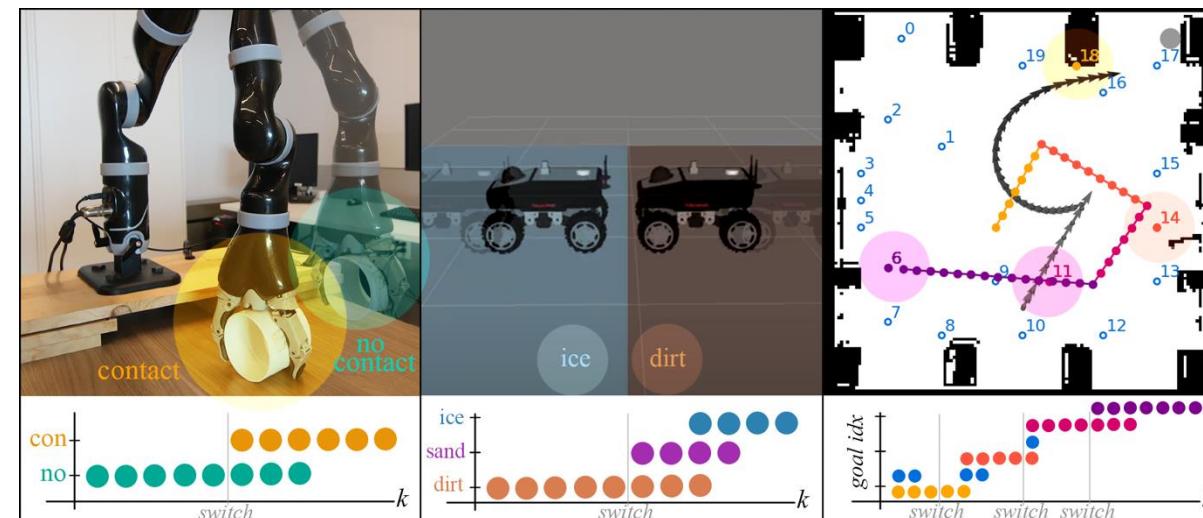
Minimize Control Effort

...by sketching before running final simulations

Sketch

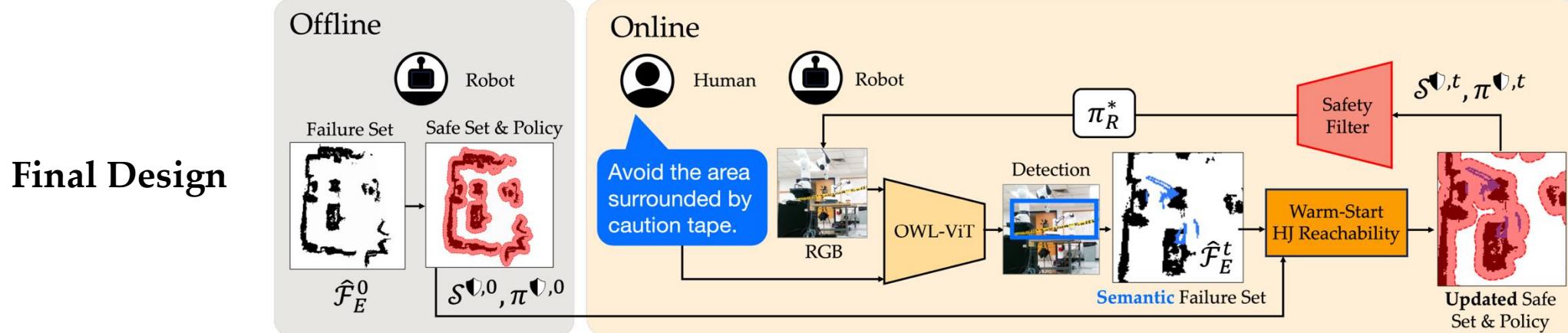
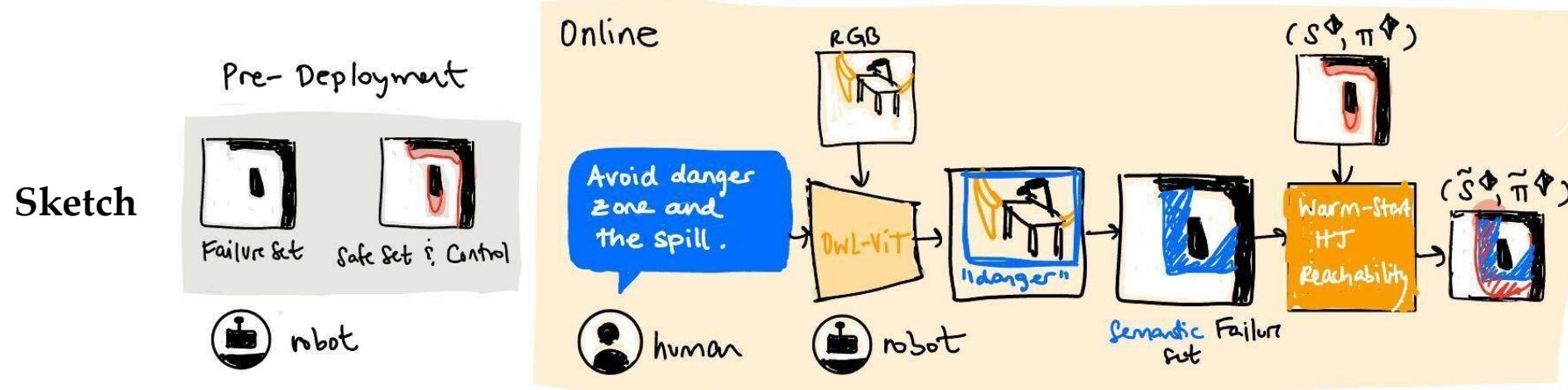


Final Design



Minimize Control Effort

...by sketching before making final figures in PPT



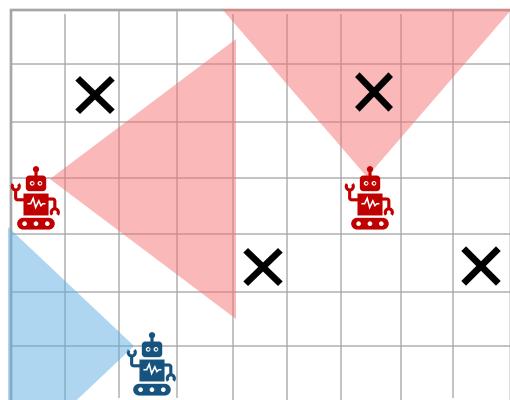
Greedy Approximation

Plan for greedily for one-step, and replan

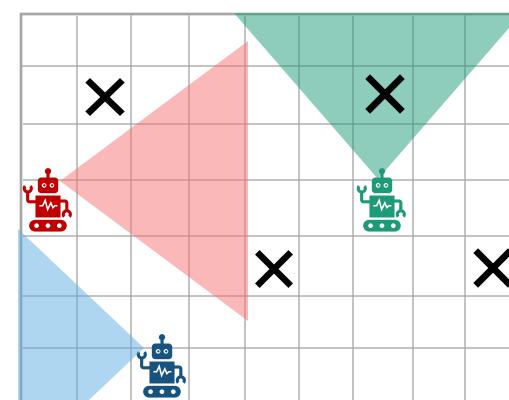
$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1)$$

$$\begin{aligned} s.t. \quad & x^1 = f(x^0, u^0) \\ & x^0 = x_{init} \end{aligned}$$

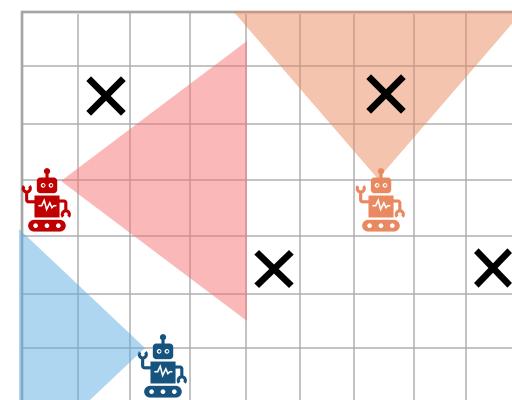
When designing, “simulate” an action and compare

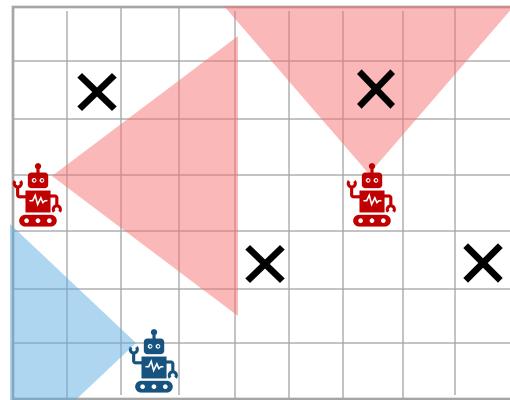


vs.

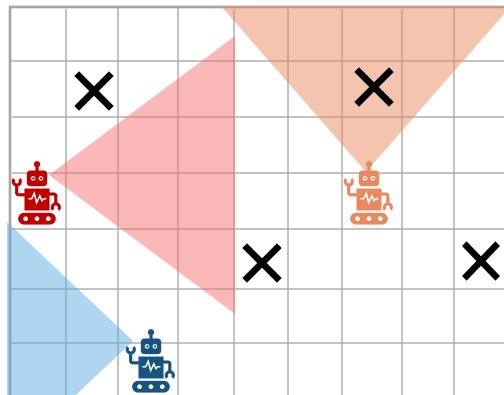


vs.

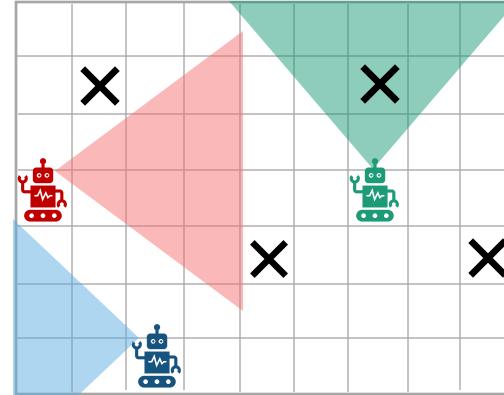



 x^0

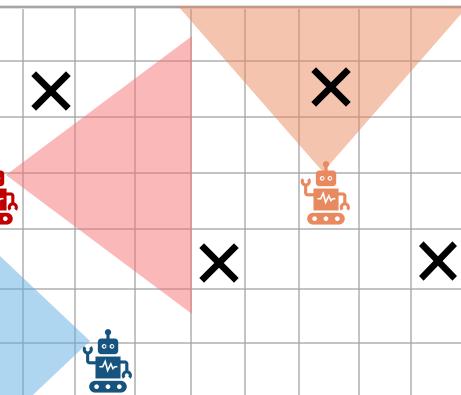
$$u = \text{color}_{gr}$$



$$x^1 = f(x^0, \text{color}_{gr})$$



$$u = \text{color}_{or}$$

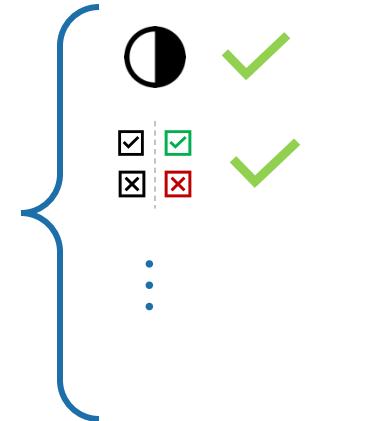


$$x^1 = f(x^0, \text{color}_{or})$$

$$x^1 = f(x^0, \text{color}_{gr})$$

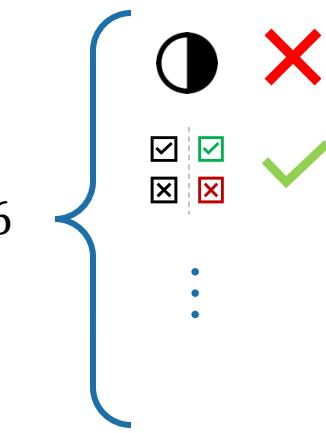
$$\text{Info}(x^1) = 5$$

$$\text{ViewEff}(x^1) = 3$$



$$\text{Info}(x^1) = 5$$

$$\text{ViewerEffort}(x^1) = 6$$



Visualizing Space & Time



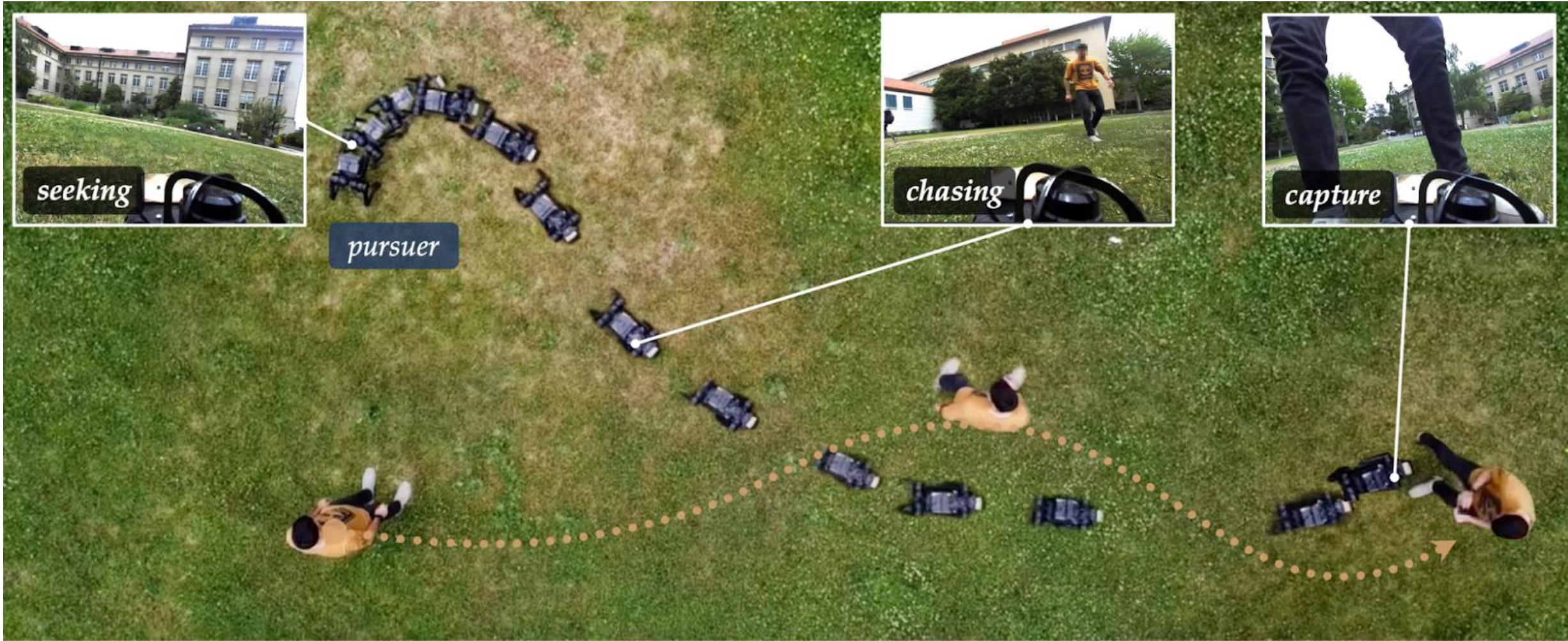
Visualizing Space & Time

...by using screenshot overlays



Visualizing Space & Time

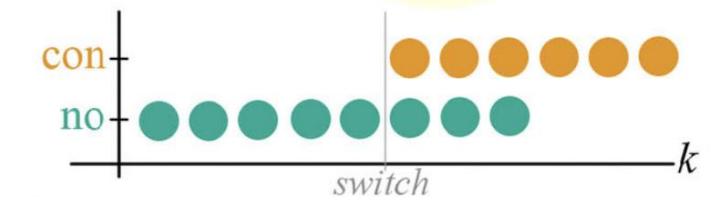
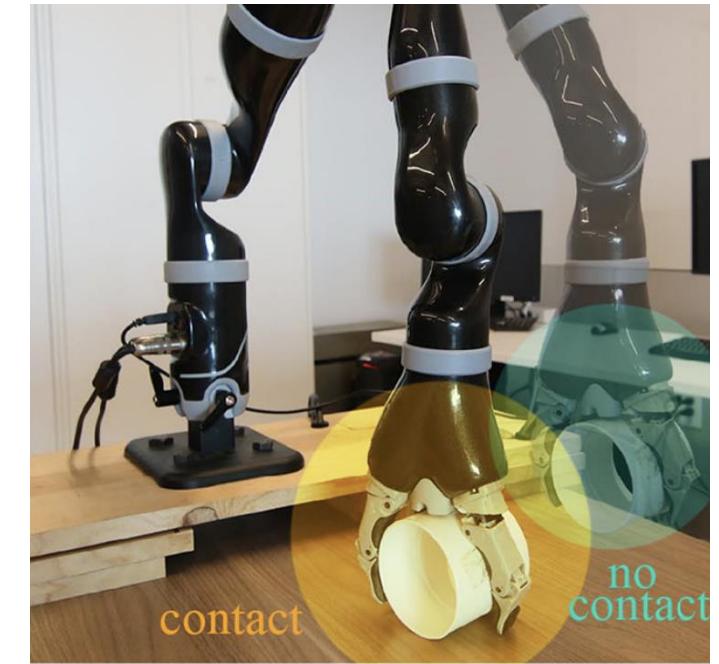
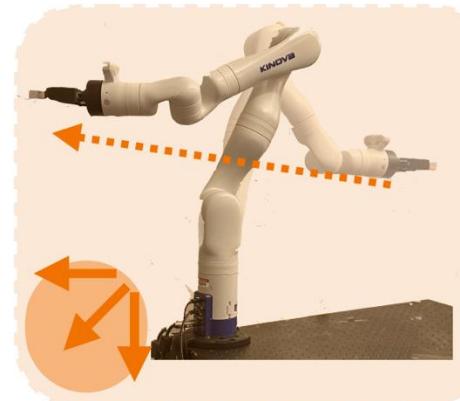
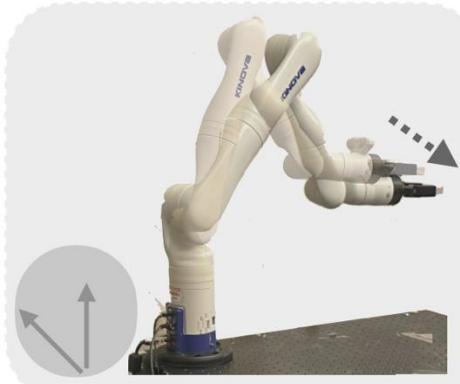
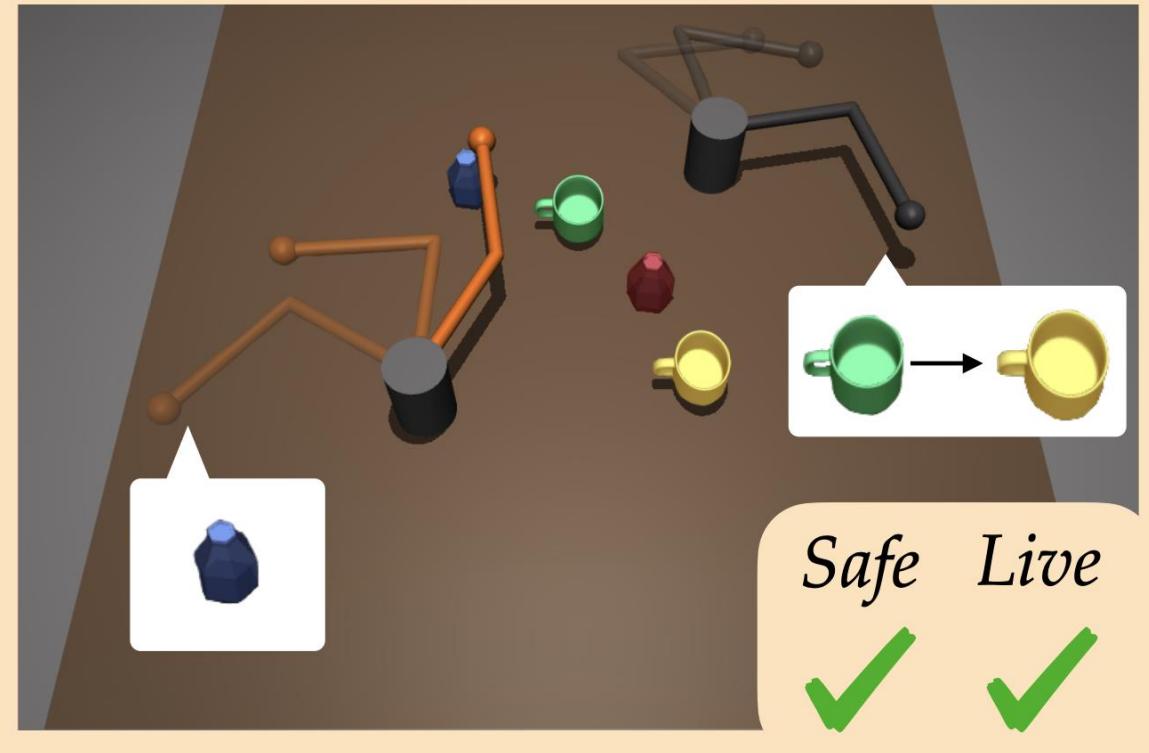
...by using screenshot overlays



Visualizing Space & Time

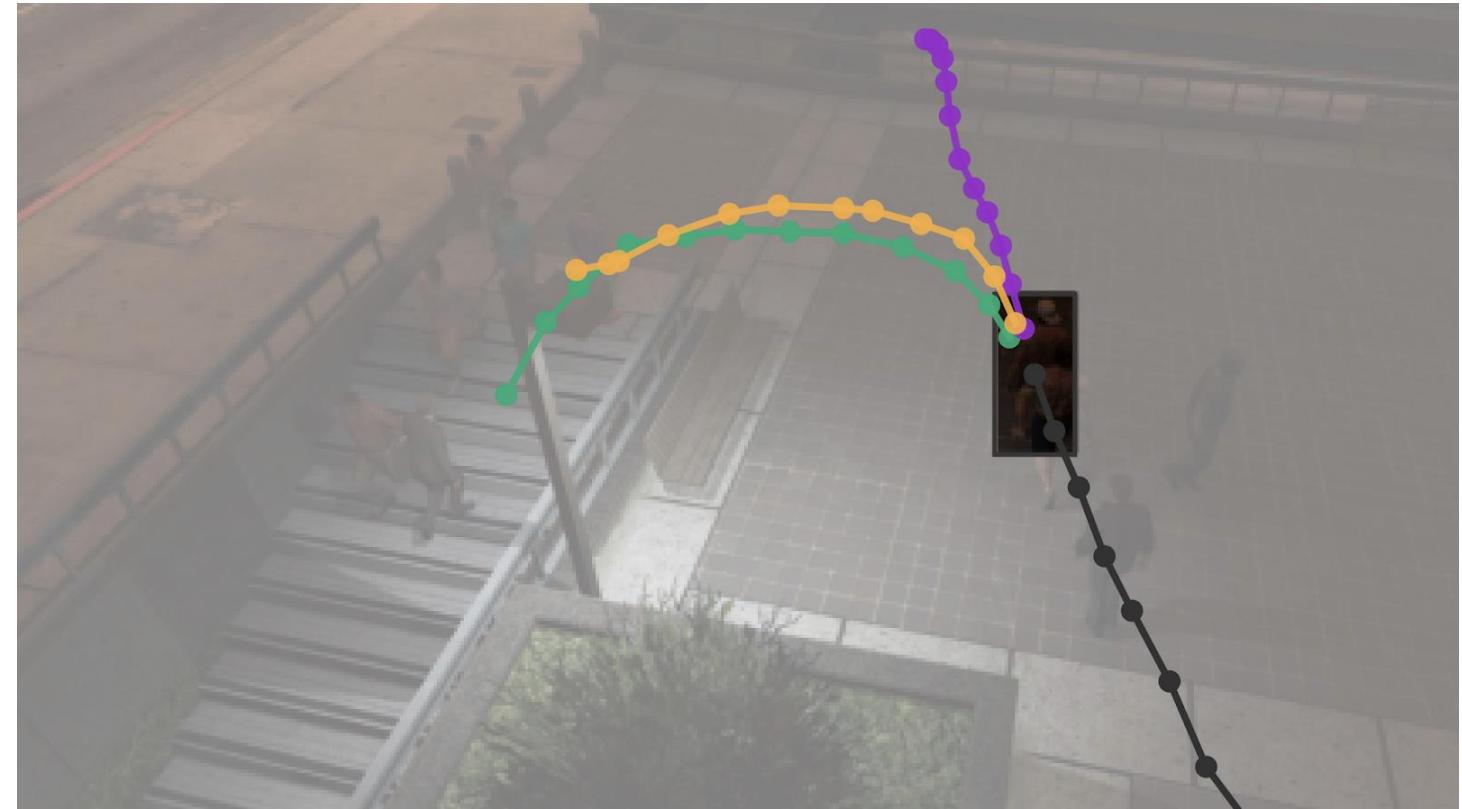
...by using transparency

SLIDE



Visualizing Space & Time

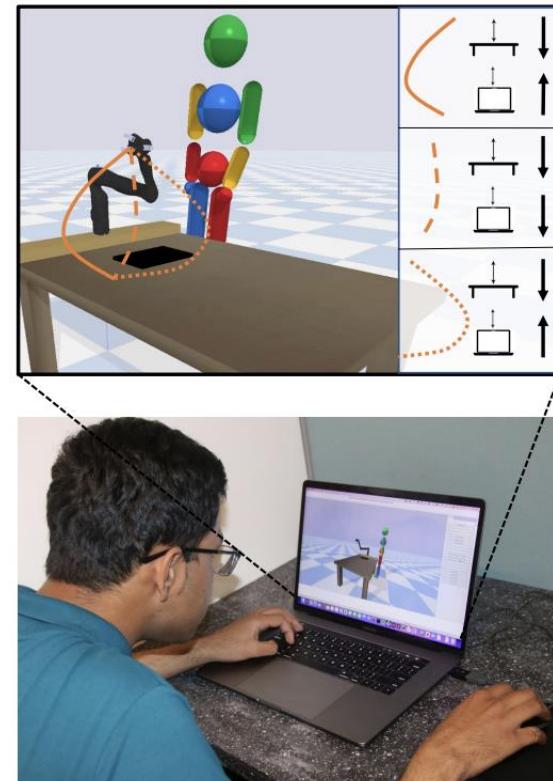
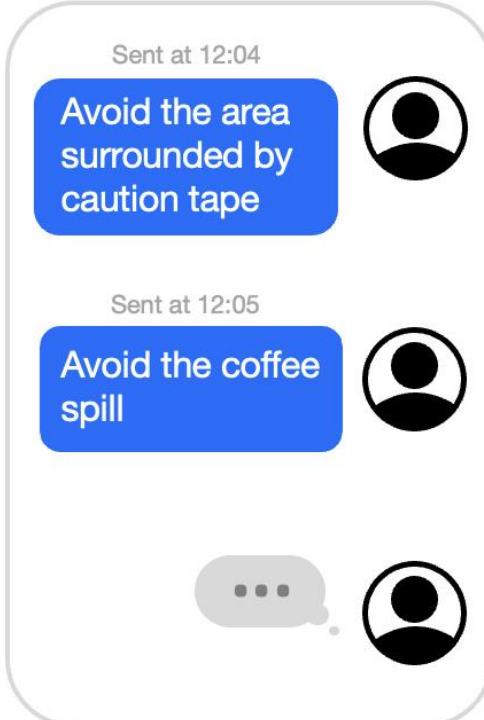
...by using colored trajectory overlays



Visualizing “human-centric” Data

... by using common iconography

Language Feedback



Bobu, A., et al. "Sirl: similarity-based implicit representation learning." HRI 2023.



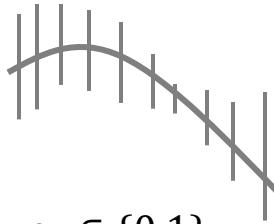
Mullen, James F., et al. "Communicating inferred goals with passive augmented reality and active haptic feedback." RA-L. (2021)

A Simple Example & Numerical Solution

"primitives"

$$e_x \in \{0,1\}$$

$$e_y \in \{0,1\}$$



$$e_A \in \{0,1\}$$

$$c_A \in \{g, o\}$$

$$p_A \in \{-, -, -\}$$

$$s_A \in \{0,1\}$$



$$e_B \in \{0,1\}$$

$$c_B \in \{g, o\}$$

$$p_B \in \{-, -, -\}$$

$$s_B \in \{0,1\}$$



$$e_{grid} \in \{0,1\}$$

state $x = [e_x, e_y, e_A, c_A, p_A, s_A, e_B, c_B, p_B, s_B, e_{grid}]$

control $u \in \{[1,0,0,0,0,0,0,0,0,0], \dots [0,0,0,0,0,0,0,0,0,1]\}$

dynamics $f(x, u) = (x + u) \% 2$

objective $\text{Info}(x) = \|x\|_2$

ViewerEffort(x) =
$$\begin{cases} +10 & \text{if } e_i = 0, i \in \{x, y, A, B\} \\ -10 & \text{if } e_i = 1 \\ -\|c_A - c_B\|_2 & \text{more different colors, less effort} \\ \|p_A - p_B\|_2 & \text{more similar the pattern, less effort} \\ \|s_A - s_B\|_2 & \text{more similar std. dev., less effort} \\ \dots & \end{cases}$$

$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1)$



optimal_designer.ipynb

File Edit View Insert Runtime Tools Help All changes saved



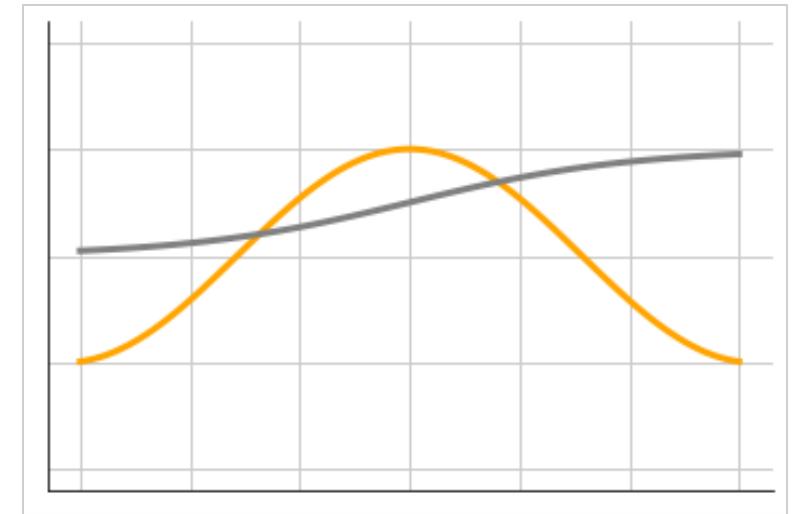
+ Code



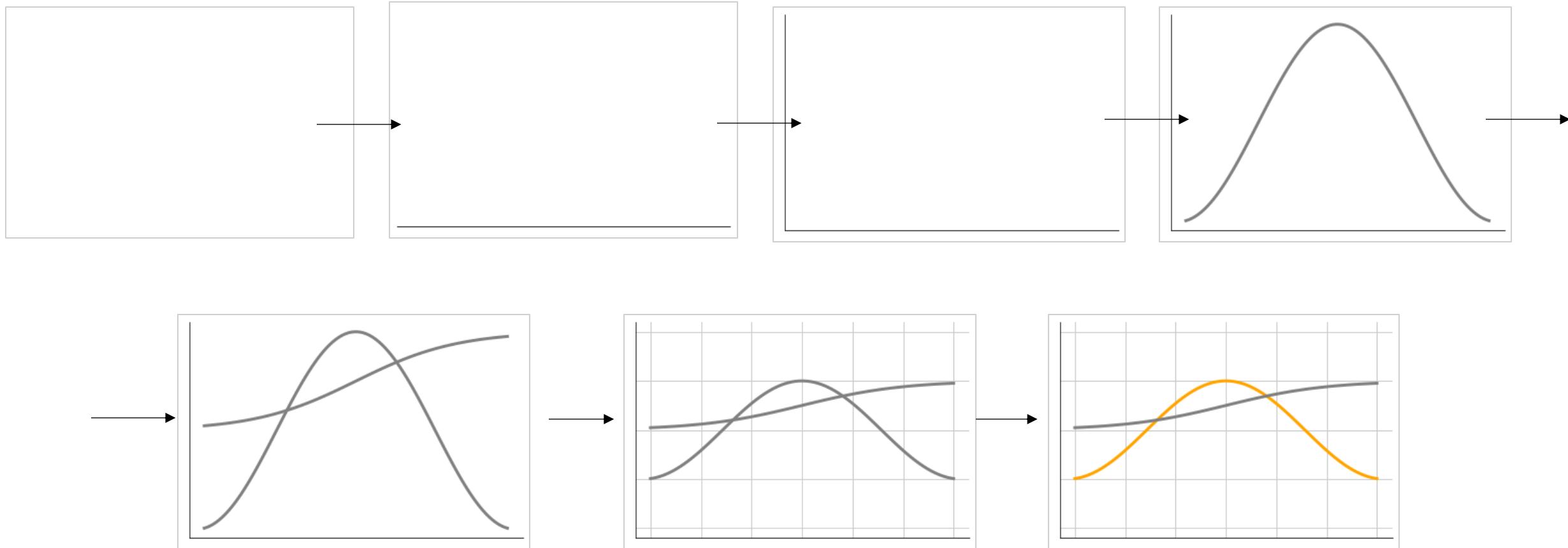
[87] 42 return r_info(x) - r_viewer_effort(x)



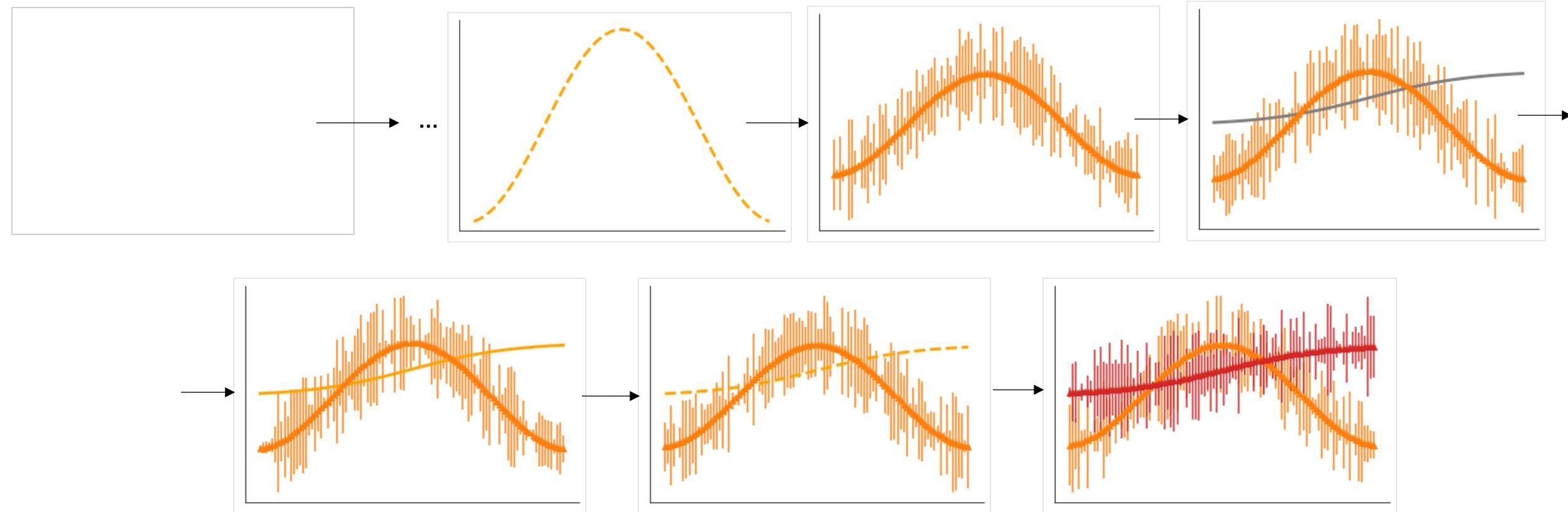
[88] 1 def get_actions(x):

2 action_list = []
3 action_list.append(np.zeros(len(x))) # add zero control
4 for i in range(len(x)):
5 u = np.zeros(len(x))
6 u[i] = 1
7 action_list.append(u)
8 return action_list[89] 1 def dynamics(x,u):
2 xnext = x + u
3 return xnext % 2[90] 1 # Solve the optimal control problem!
2 def optimal_designer(x0, T):
3 x = x0
4 xtraj = []
5 xtraj.append(x)
6
7 for t in range(T):
8 uopt = np.zeros(len(x)) # zero control
9 for u in get_actions(x):
10 xnext = dynamics(x,u)
11 r = reward(xnext)
12 if r > reward(dynamics(x,uopt)):
13 uopt = u
14 # execute optimal control
15 x = dynamics(x,uopt)
16 xtraj.append(x)
17
18 return xtraj*The Optimal Design!*

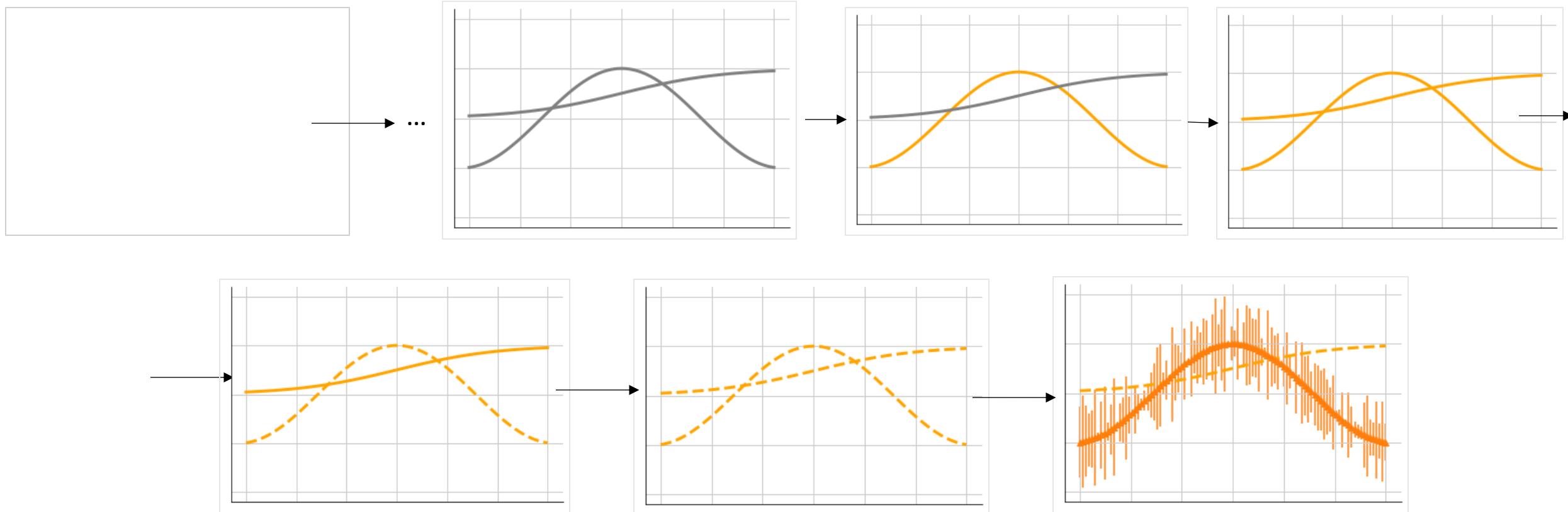
The Optimal Design Trajectory: $\text{Info}(x^T) - \text{ViewerEffort}(x^T)$



If you only care about $\text{Info}(x^T)$...



If you care more about Info: $100 \cdot \text{Info}(x^T) - \text{ViewerEffort}(x^T)$...



Front Figures

$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1)$$

$$\begin{aligned} s.t. \quad & x^1 = f(x^0, u^0) \\ & x^0 = x_{init} \end{aligned}$$

*Communicate your contribution.
Show what makes your work “impactful”?*

$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1) + \text{Impact}(x^1)$$

$$\begin{aligned} s.t. \quad & x^1 = f(x^0, u^0) \\ & x^0 = x_{init} \end{aligned}$$



Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu

Abstract—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of creating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. Therefore, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, relevant, and diverse. We use these principles to formalize augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on

ata to enable more data-efficient online learning.

Paper from RSS 2022

I. INTRODUCTION



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative model, which is hard to find, and requires a large amount of labeled data to learn the model well.

Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu

Abstract—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of creating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. Therefore, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, relevant, and diverse. We use these principles to formalize augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

I. INTRODUCTION



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative model, which is hard to find, and requires a large amount of labeled data.



Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu

 **Abstract**—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of creating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. Therefore, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, relevant, and diverse. We use these principles to formalize augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

I. INTRODUCTION



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative model, which is hard to find, and requires a large amount of labeled data to learn the model well.

Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

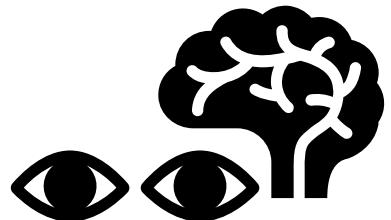
Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu

Abstract—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and therefore learning from them is a challenging problem. Within computer vision, one way of dealing with the lack of data is *data augmentation*, which creates additional training data by applying transformations to existing data. However, because the data used in deep learning is often very different from the data used in computer vision, this approach is not always effective.

Therefore, we propose a new method for data augmentation in robotic manipulation. We argue that the data used in deep learning for manipulation is relevant, and diverse. We use these principles to formalize data augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to the trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.



Brain expects a relationship between title, abstract, front fig

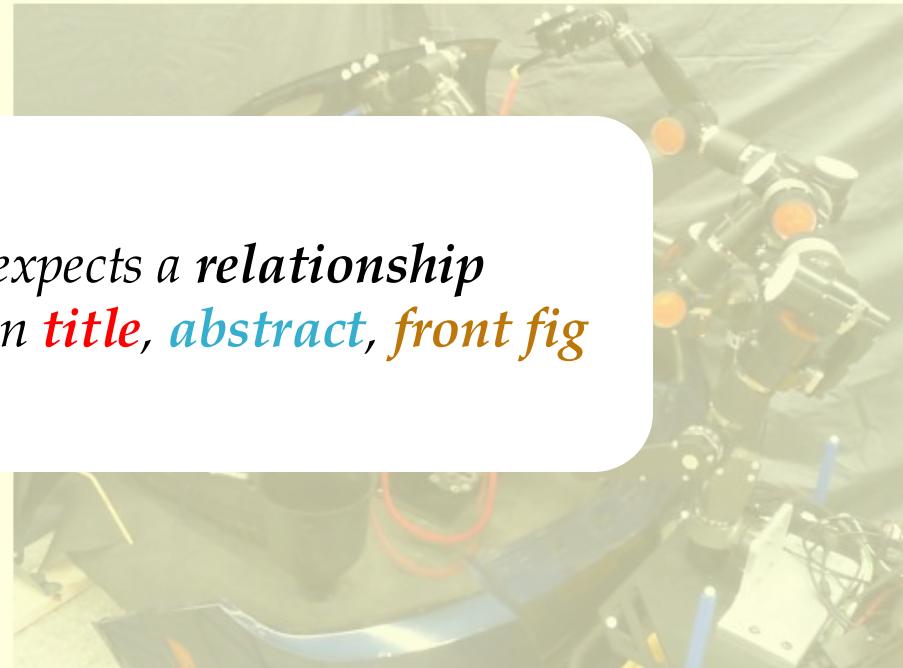


Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

Data Augmentation for Manipulation

Peter Mitrano

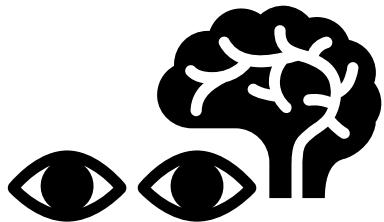
University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu



$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1) + \text{Impact}(x^1)$$

*Impact is increased if
the relationship is clear*

domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

I. INTRODUCTION



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative model, which is often not the case in real-world scenarios.

Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu

Abstract—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation

any learning problems for which such datasets do not exist are challenging. Collecting these datasets is time-consuming and expensive, while learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of generating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. In this paper, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, diverse, and diverse. We use these principles to formalize data augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation task. This method applies rigid body transformations to both geometric state and action data. We test our method on two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks.

We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

- + Shows real robot system
- Bg distracts
- No insight into approach
- Disconnect from title



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

I. INTRODUCTION

model [19, 15, 2]. This approach assumes a useful generative model, such as a learned function, that is able to sample from

Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

Email: dmitryb@umich.edu

Abstract—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation

any learning problems for which such datasets do not exist creating these datasets is time-consuming and expensive, therefore learning from small datasets is an important open problem within computer vision, a common approach to a lack of data augmentation. Data augmentation is the process of generating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods developed for computer vision cannot be easily adapted to manipulation. In this work, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, diverse, and diverse. We use these principles to formalize data augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation task. This method applies rigid body transformations to both geometric state and action data. We test our method on two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks.

We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

+ Shows real robot system

+ Lightened bg

- No insight into approach

- Disconnect from title

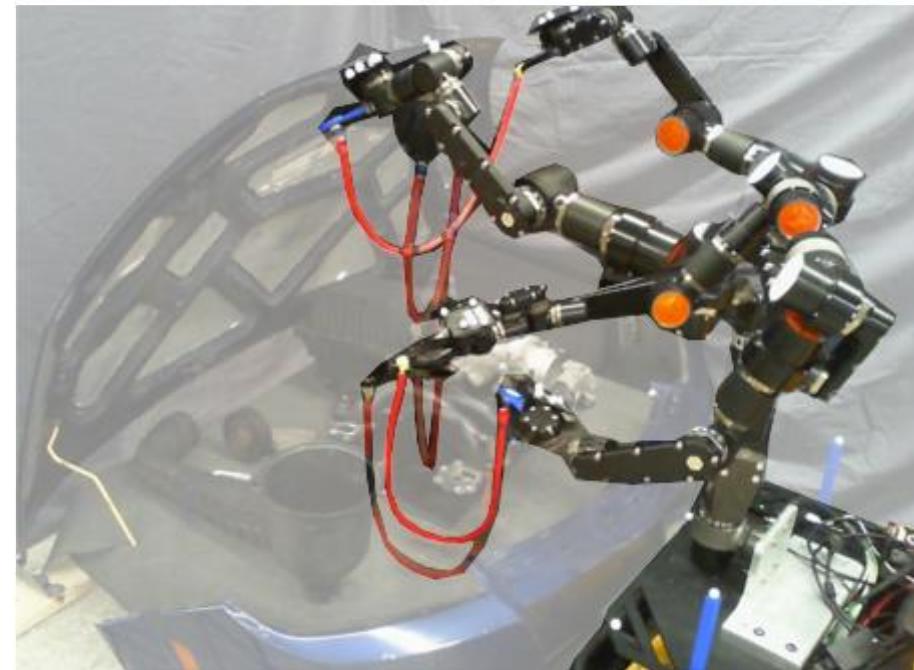


Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

Data Augmentation for Manipulation

Peter Mitrano

University of Michigan

Email: pmitrano@umich.edu

Dmitry Berenson

University of Michigan

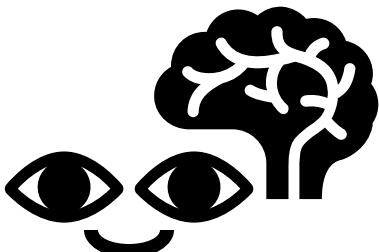
Email: dmitryb@umich.edu

Abstract—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation any learning problems for which such datasets do not exist collecting these datasets is time-consuming and expensive, where learning from small datasets is an important open problem within computer vision, a common approach to a lack of data augmentation. Data augmentation is the process of generating additional training examples by modifying existing ones. Because the types of tasks and data differ, the methods from computer vision cannot be easily adapted to manipulation. We propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, diverse, and generalizable. We use these principles to formalize data augmentation as an optimization problem, with the objective of maximizing the information gain from physics and knowledge of the manipulation method. Our method applies rigid body transformations to both geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar objects in a car engine bay, and 2) learning a constraint checker for a robotic manipulator. These two scenarios have different data requirements yet in both scenarios, training on our augmented dataset improves performance on downstream tasks. Finally, we show how our augmentation method can be used on its own to enable more data-efficient online learning.

+ Shows real robot system

+ Lightened bg

+ More insight



Relationship between title, abstract, front fig

I. INTRODUCTION

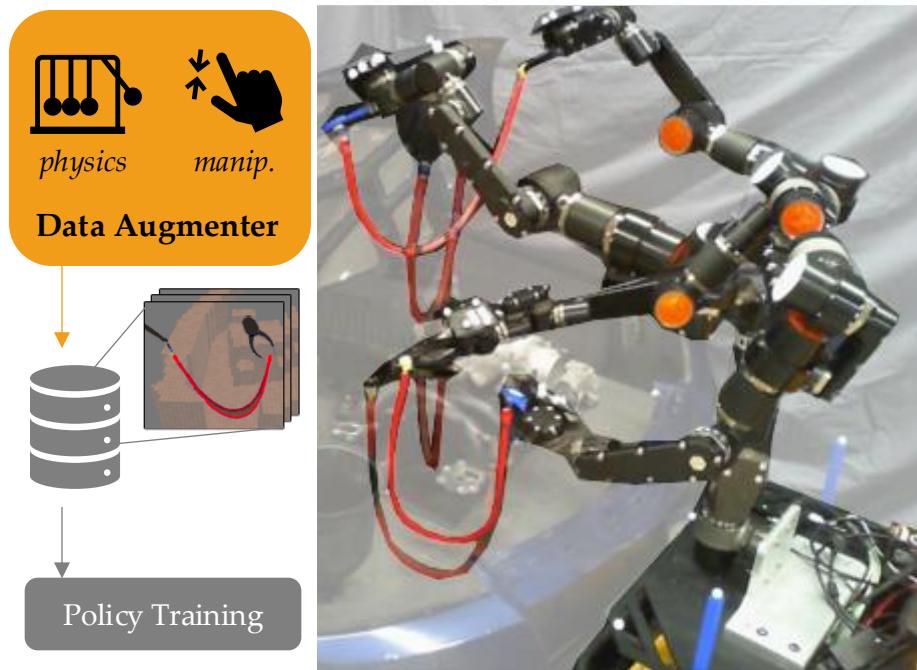


Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative model, such as a learned function approximation, that can sample from the space of possible actions. If such a model is available, then the data augmentation process can be designed to focus on improving the quality of the generated samples, rather than simply increasing their quantity. This can lead to more efficient and effective learning, especially for tasks that require complex, multi-step reasoning or decision-making.

A Learning-based Iterative Control Framework for Controlling a Robot Arm with Pneumatic Artificial Muscles

Hao Ma, Dieter Büchler, Bernhard Schölkopf, Michael Muehlebach

- + Shows real robot system
- + Bg removed for clarity
- Irrelevant annotations (?)
- No insight into approach
- Disconnect from title

have been widely used for various tasks, such as aerospace [32], medical applications, [26, 22], and industry, [11, 31]. Due to their high power-to-weight ratio [11], PAMs have been used in [21]. Moreover, due to their low cost and high reliability, PAMs have been widely used in various applications, such as aerospace [32], medical applications, [26, 22], and industry, [11, 31]. Due to their high power-to-weight ratio [11], PAMs have been used in [21].

In this work, we propose a new learning-based iterative control (IC) framework that enables a complex soft-robot to track trajectories accurately. Compared to traditional learning control (ILC), which operates on a single reference trajectory, we use a deep learning approach to learn across various reference trajectories. The resulting policy computes feedforward actions and is used in a hybrid control design. Our method incorporates knowledge about the system dynamics and by learning feedforward actions, it mitigates the risk of instability. We demonstrate low sample complexity and an excellent tracking performance in real-world experiments. The experiments are conducted on a custom-made robot arm with four degrees of freedom actuated with pneumatic artificial muscles. The robot arm is capable of performing high acceleration and high velocity motion.

I. INTRODUCTION

Pneumatic artificial muscles (PAMs) have been widely used for various tasks, such as aerospace [32], medical applications, [26, 22], and industry, [11, 31]. Due to their high power-to-weight ratio [11], PAMs have been used in [21]. Moreover, due to their low cost and high reliability, PAMs have been widely used in various applications, such as aerospace [32], medical applications, [26, 22], and industry, [11, 31]. Due to their high power-to-weight ratio [11], PAMs have been used in [21].

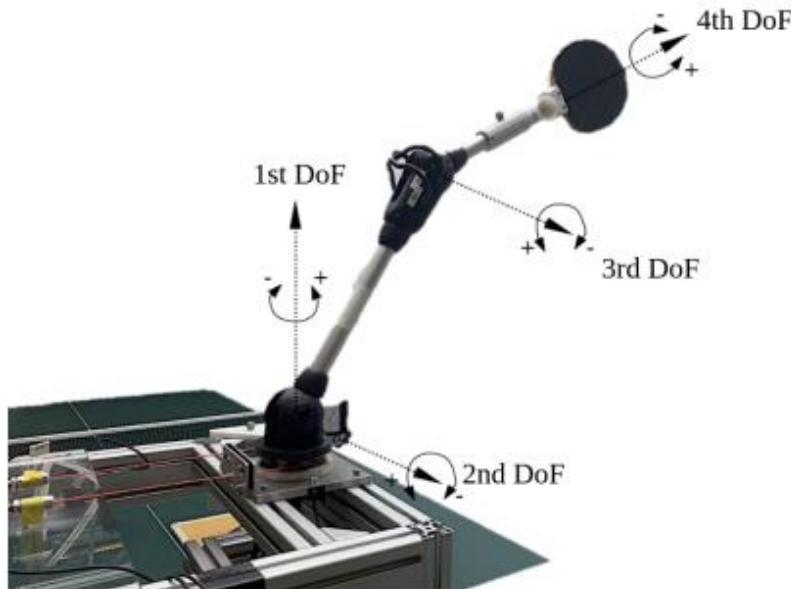


Fig. 1: The figure shows the structure of the robot arm. It has four rotational joints, and each joint is actuated by a pair of PAMs. For simplicity we consider only the first three degrees of freedom in this article. Note that DoF stands for degree of freedom.

Certifiable Robot Design Optimization using Differentiable Programming

Charles Dawson

*Dept. of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, USA
cbd@mit.edu*

Chuchu Fan

*Dept. of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, USA
chuchu@mit.edu*



Figure shows framework



Hard to read



No insight into approach



Poor image cropping

-There is a growing need for computational tools to automatically design and verify autonomous systems, especially robotic systems involving perception, planning, and hardware in the autonomy stack. Differentiable programming has recently emerged as a powerful tool for modeling optimization. However, very few studies have been done on how differentiable programming can be used for reliable end-to-end design optimization. In this paper, we fill this gap by combining differentiable programming for design optimization with a novel statistical framework for certifying the robustness of optimized designs. Our framework enables end-to-end optimization and robustness certification for complex systems, enabling simultaneous optimization of navigation, perception, planning, control, and hardware subsystems.

Using simulation and hardware experiments, we show how our tool can be used to solve practical problems in robotics. First, we demonstrate how to find optimal placements for robot navigation (a design with 10 tunable parameters) and 6 tunable parameters) in under 5 minutes to achieve an 0.4x performance improvement compared to the initial

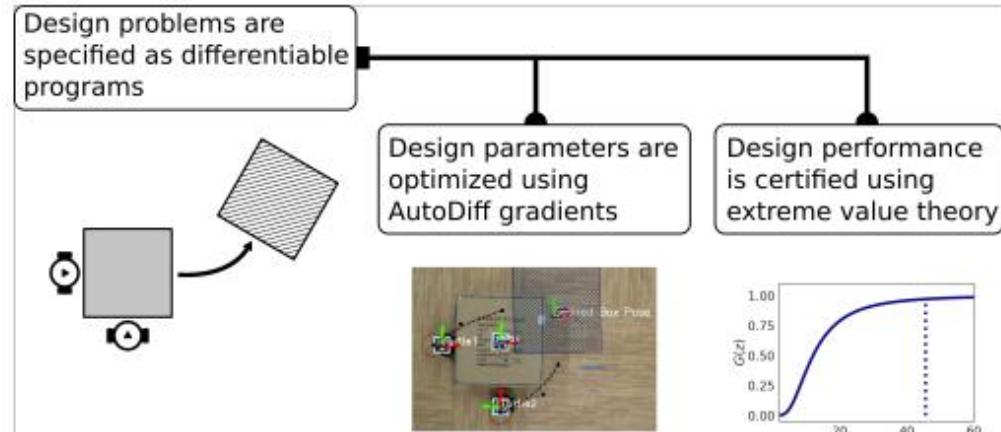


Fig. 1: An overview of our framework for robot design optimization and certification. Differentiable programming allows the user to flexibly specify a robot design problem, which can be efficiently optimized using exact gradients and verified using an extreme value statistical analysis.

Learning Category-Specific Mesh Reconstruction from Image Collections

Angjoo Kanazawa*, Shubham Tulsiani*, Alexei A. Efros, Jitendra Malik

University of California, Berkeley

{kanazawa, shubhtuls, efros, malik}@eecs.berkeley.edu

- + Shows real data / application
- + Visualizes methodology
- + Connected to title

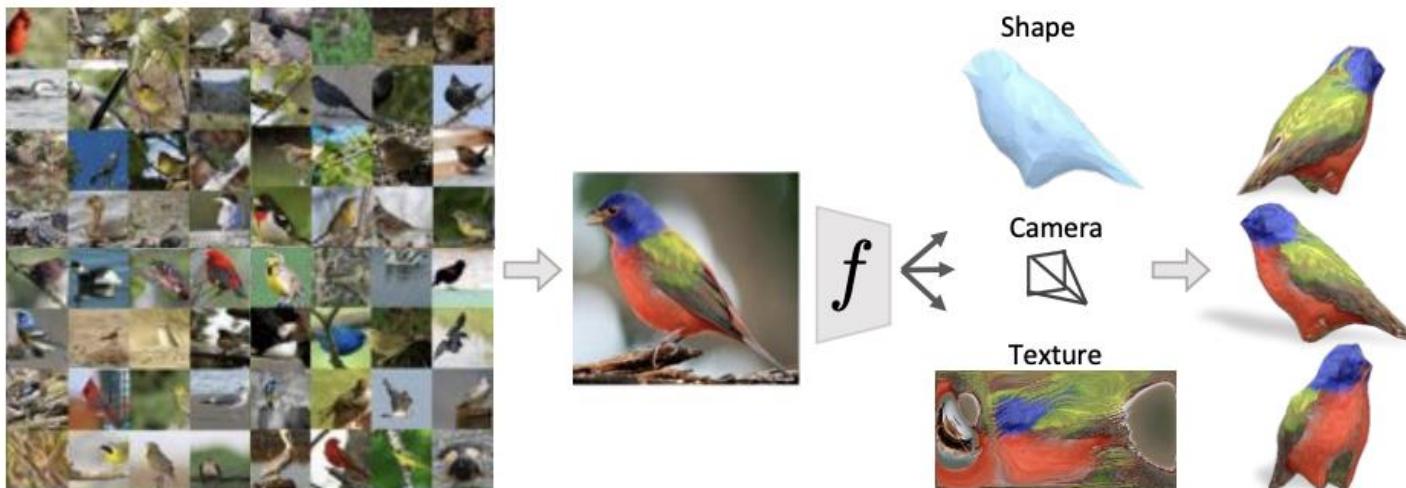
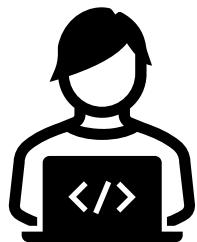


Fig. 1: Given an annotated image collection of an object category, we learn a predictor f that can map a novel image I to its 3D shape, camera pose, and texture.

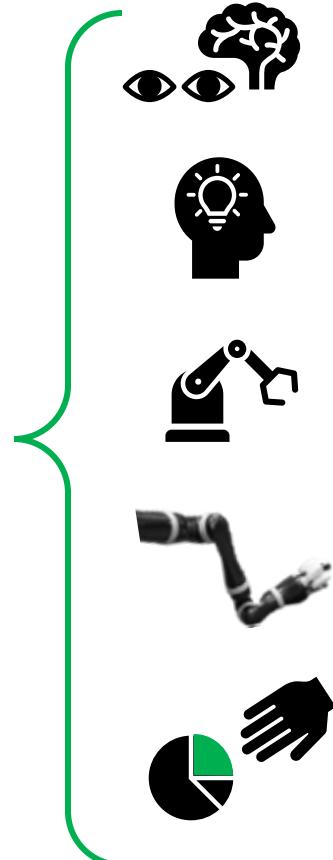
How do we design this reward function?

$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1) + \text{Impact}(x^1)$$



You
(the designer)

high Impact



Clear relationship between title, abstract, front fig

Give insight into your approach (e.g., why/how it works)

Use cartoons for “framework” illustrations...

..but also showcase real robots /data if you have it

Highlight (and/or contextualize) your contribution

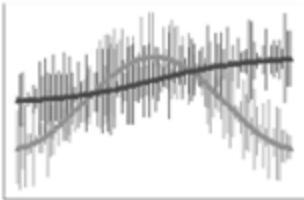


The viewer

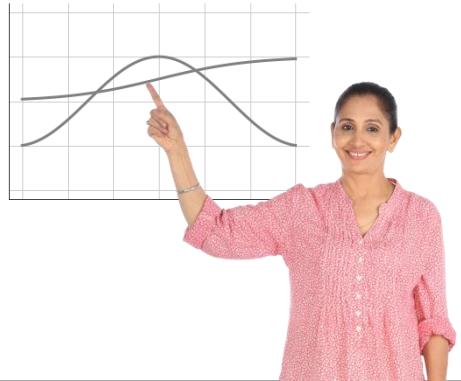
From theory to practice: Slides

Good news – the same principles apply!

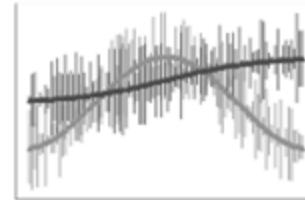
- Increased information
- Increased viewer effort



- Increased information
- Decreased viewer effort



ℹ️ ↑ ☺️ ↓ ℹ️ ↑ ☺️ ↑



focus on slide

icons > text

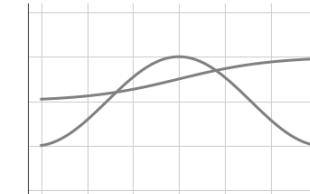
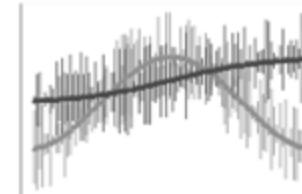
focus on speaker

If I showed you this slide sequence...

$$\begin{aligned} \max_{u^{0:T}} \quad & \text{Info}(x^T) - \text{ViewerEffort}(x^T) \\ \text{s.t.} \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{init} \end{aligned}$$



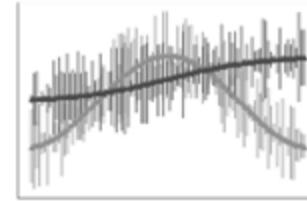
i ↑ ☺ ↓ **i** ↑ ☺ ↑



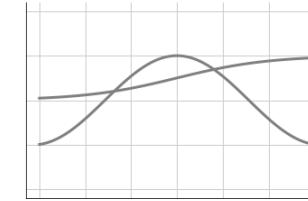
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\begin{aligned} s.t. \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{init} \end{aligned}$$



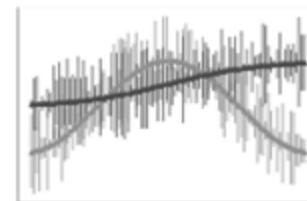
   



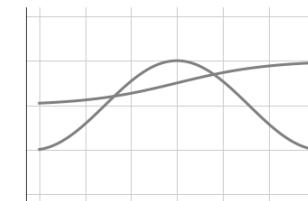
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\begin{aligned} s.t. \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{init} \end{aligned}$$

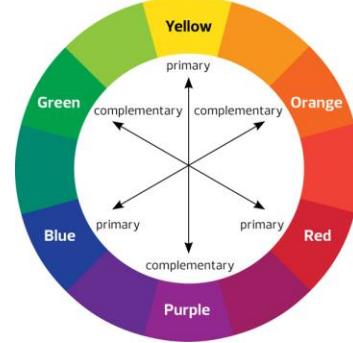




use colors to encode meaning

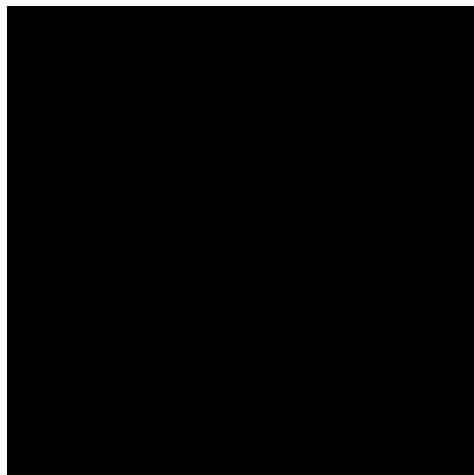
color “rules”



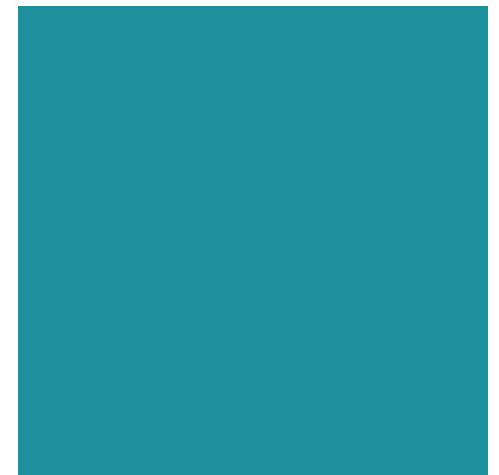
start w/ primary color

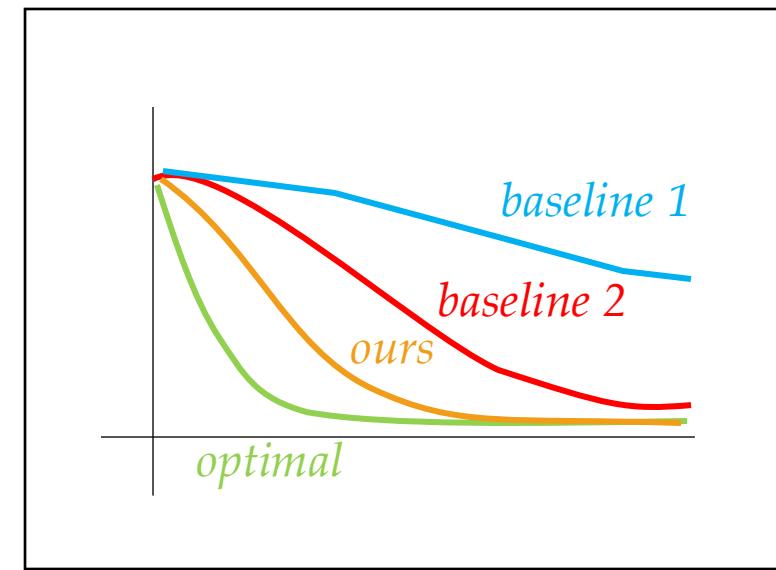
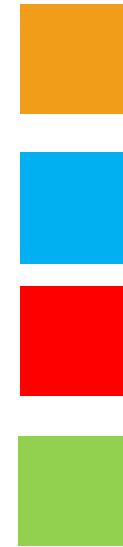
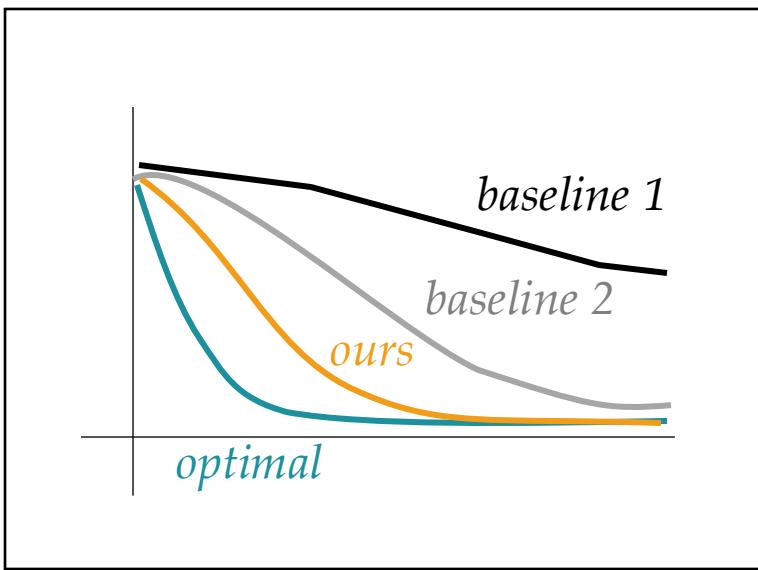
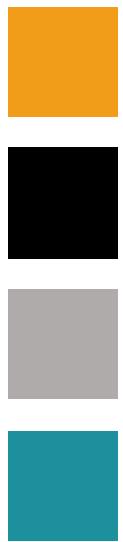


neutral colors



accent color





$$\begin{aligned} \max_{\mathbf{u}^{0:T}} \quad & \text{Info}(x^T) - \text{ViewerEffort}(x^T) \\ \text{s.t.} \quad & x^{t+1} = f(x^t, \mathbf{u}^t) \\ & x^0 = \mathbf{x}_{init} \end{aligned}$$

less visible colors

$$\begin{aligned} \max_{\mathbf{u}^{0:T}} \quad & \text{Info}(x^T) - \text{ViewerEffort}(x^T) \\ \text{s.t.} \quad & x^{t+1} = f(x^t, \mathbf{u}^t) \\ & x^0 = \mathbf{x}_{init} \end{aligned}$$

*more visible colors**

**subject to projector / TV quality and lighting conditions*

$$\max_{\mathbf{u}^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$s.t. \quad \begin{aligned} x^{t+1} &= f(x^t, \mathbf{u}^t) \\ x^0 &= x_{init} \end{aligned}$$



focus on slide

$$\max_{\mathbf{u}^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$s.t. \quad \begin{aligned} x^{t+1} &= f(x^t, \mathbf{u}^t) \\ x^0 &= x_{init} \end{aligned}$$



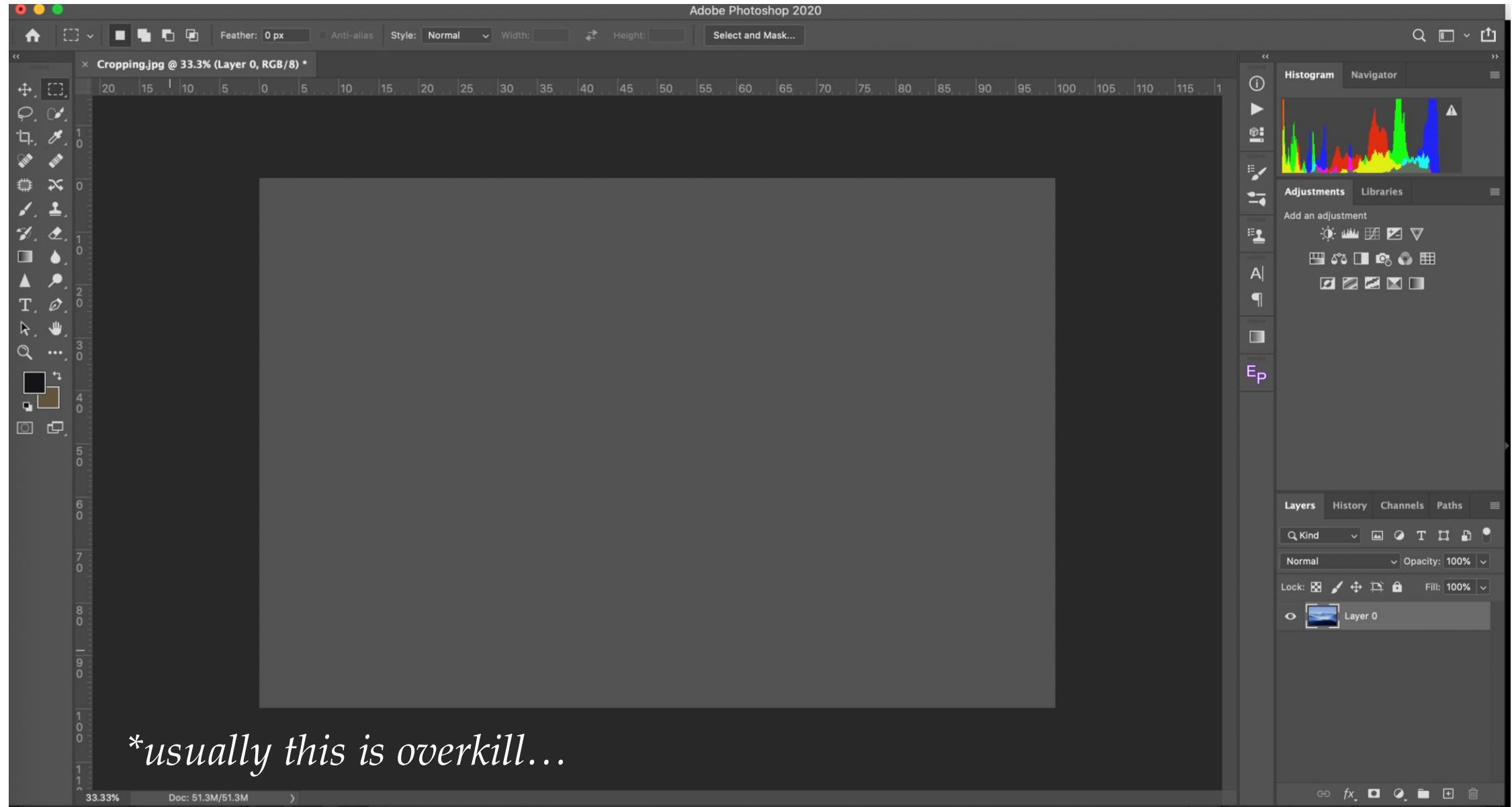
focus on speaker

Where to get

*tools
figures
fonts
color schemes*

Tools

Photoshop*



*usually this is overkill...

Tools

Free Web-browser-based “Photoshop”

The screenshot shows the homepage of Photopea, a free web-based image editor. The interface is dark-themed with white text and icons. At the top, there's a navigation bar with links for File, Edit, Image, Layer, Select, Filter, View, Window, More, Account (which is highlighted in red), About, Report a bug, Learn, Blog, API, and social media links for Reddit, Twitter, and Facebook. A message "Ad blocking detected" is visible near the top right. On the left, a sidebar lists options: Home (selected), This Device, Google Drive, Shared Drives, Private Folder, OneDrive, and Dropbox. The main area features the Photopea logo and the word "Photopea" in large white letters. Below it are three buttons: "★ New Project", "Open From Computer", and "PSD Templates". There's a large central area with a placeholder text "Drop any files here". At the bottom, there are icons for various file formats: PSD, AI, XD, .sketch, PDF, XCF, RAW, and ANY, each with its extension below it.

photopea.com

File Edit Image Layer Select Filter View Window More Account About Report a bug Learn Blog API Ad blocking detected

Home This Device Google Drive Shared Drives Private Folder OneDrive Dropbox

★ New Project Open From Computer PSD Templates

Drop any files here

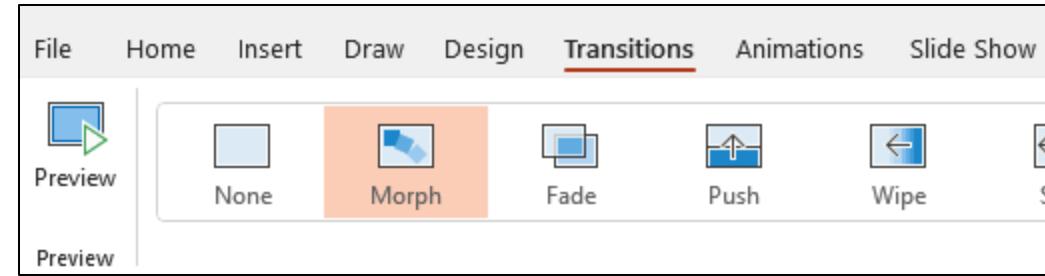
.PSD .AI .XD .sketch .PDF .XCF RAW ANY

fiverr. Give your business the perfect logo. Get Yours Custom-Made

Tools

Don't underestimate the power in PowerPoint ☺

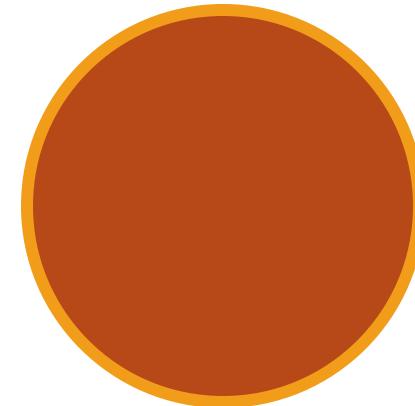
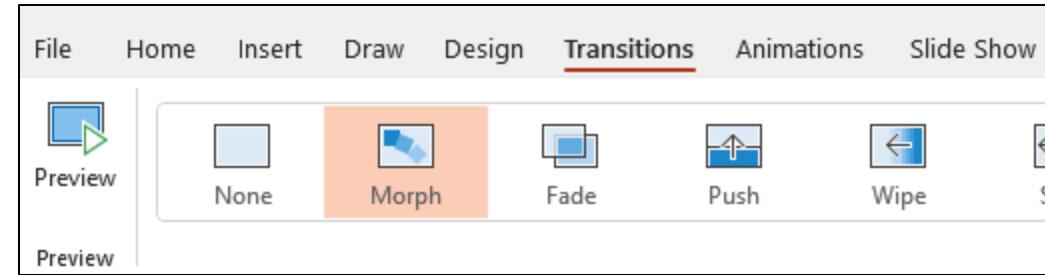
Morph Transition
*makes everything look like you
spent a lot of time animating it*



Tools

Don't underestimate the power in PowerPoint ☺

Morph Transition
*makes everything look like you
spent a lot of time animating it*



Tools

Don't underestimate the power in PowerPoint ☺

Don't underestimate the power in PowerPoint ☺

Morph Transition makes everything look like you spent a lot of time animating it

Tools

Slide 1

Don't underestimate the power in PowerPoint ☺

Morph Transition makes everything look like you spent a lot of time animating it

Tools

Copy of Slide 1 + change to circle

Tools

Don't underestimate the power in PowerPoint ☺

Really good practice to animate equations “step by step”

$$\dot{x} = f(x, u)$$

Tools

Don't underestimate the power in PowerPoint ☺

Really good practice to animate equations “step by step”

$$\dot{x} = f(x, \textcolor{red}{u}, \textcolor{blue}{d})$$

Tools

Don't underestimate the power in PowerPoint ☺

Really good practice to animate equations “step by step”

$$\nabla_x V(x, t) \cdot f(x, \textcolor{red}{u}, \textcolor{blue}{d})$$

Tools

Don't underestimate the power in PowerPoint ☺

Really good practice to animate equations “step by step”

$$\max_{\textcolor{red}{u}} \min_{\textcolor{blue}{d}} \nabla_x V(x, t) \cdot f(x, \textcolor{red}{u}, \textcolor{blue}{d})$$

Tools

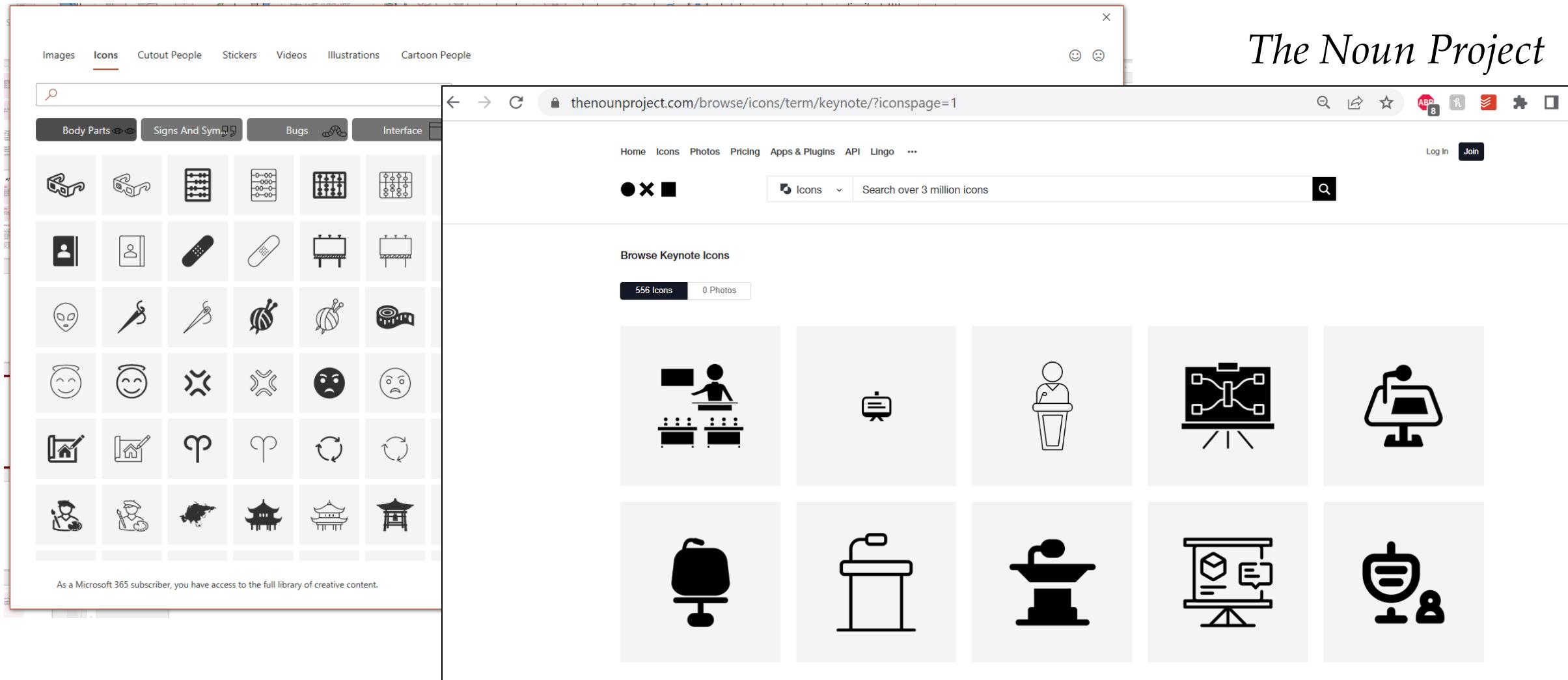
Don't underestimate the power in PowerPoint ☺

Really good practice to animate equations “step by step”

$$\max_{\textcolor{red}{u}} \min_{\textcolor{blue}{d}} \nabla_x V(x, t) \cdot f(x, \textcolor{red}{u}, \textcolor{blue}{d}) + \frac{\partial V}{\partial t} = 0$$

Figures

PowerPoint Icons



Fonts

Google Fonts

The screenshot shows the Google Fonts homepage. At the top, there's a search bar with 'Search fonts' and a sentence input field with 'Type something'. A font size slider is set to '40px'. Below the search area are filters for 'Categories', 'Language', 'Font properties', and a checkbox for 'Show only variable fonts'. The main content area displays a grid of font families. Each card includes the font name, author, number of styles, and a preview of the text 'Almost before we knew it, we had left the ground.'.

| Font Family | Author | Number of Styles | Preview Text |
|--------------------|---------------------------|------------------|---|
| Roboto | Christian Robertson | 12 styles | Almost before we knew it, we had left the ground. |
| Joan | Paolo Biagini | 1 style | Almost before we knew it, we had left the ground. |
| Nuosu SIL | SIL International | 1 style | Almost before we knew it, we had left the ground. |
| Kdam Thmor Pro | Tep Sovichet, Hak Longdey | 1 style | Almost before we knew it, we had left the |
| Open Sans | Steve Matteson | Variable | Almost before we knew it, we had left |
| Noto Sans Japanese | Google | 6 styles | 人類社会のすべての構成員の固有の尊厳と平 |

Colors

learnui.design/tools/color-picker.html#palette

DATA COLOR PICKER powered by LEARN UI DESIGN

UI COURSE UX COURSE BLOG TOOLS DESIGN HACKS

PALETTE SINGLE HUE DIVERGENT

PALETTE GENERATOR

NUMBER OF COLORS 3 4 5 6 7 8

BACKGROUND COLOR

#003f5c #58508d #bc5090 #ff6361 #ffa600

Color palette designers

DATA COLOR PICKER powered by LEARN UI DESIGN

UI COURSE UX COURSE BLOG TOOLS

PALETTE SINGLE HUE DIVERGENT

DIVERGENT COLOR SCALE

NUMBER OF COLORS 5 7 9 11 13

MODIFY MIDPOINT COLOR BRIGHTNESS COLOR INTENSITY

BACKGROUND COLOR

#ff9500 #ffb365 #ffd2aa #f1f1f1 #bbdcea #7ec7e2 #00b3db

Colors

 ACCESSIBLE COLOR GENERATOR powered by [LEARN UI DESIGN](#)

Show me the closest variations of  [#4ac4e2](#)

that contrast against  [the color](#)  [#ffffff](#)

enough to meet [AA Guidelines](#) 

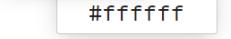
RESULTS

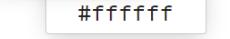
FOR LARGE/BOLD TEXT  Try this combo instead:


[#06a2bf](#)

FOR SMALL TEXT  Try this combo instead:

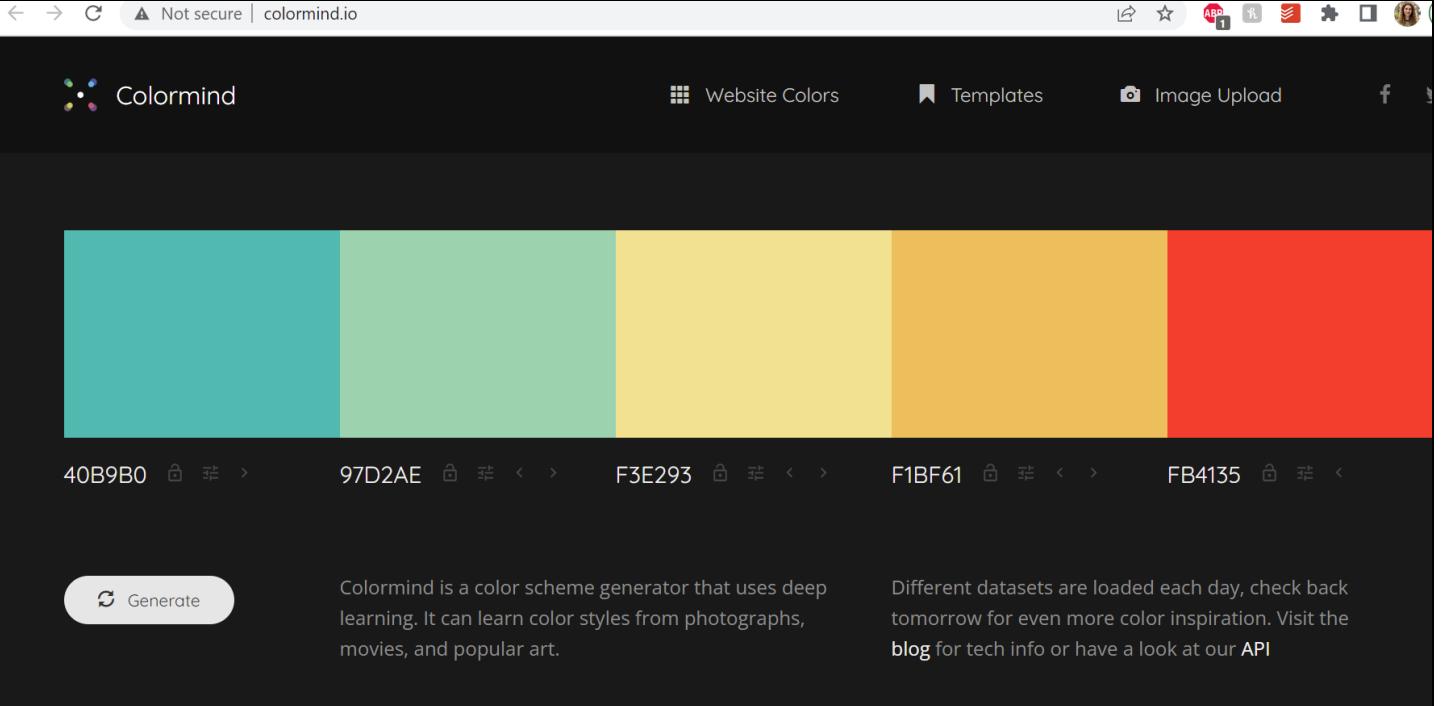

[#00819d](#)


[#ffffff](#)


[#ffffff](#)

Accessible colors

Deep-learning-generated palettes 😊



The screenshot shows the Colormind website interface. At the top, there's a navigation bar with links for "Website Colors", "Templates", and "Image Upload". Below the navigation, a row of five color swatches is displayed, each with a hex code and a "Generate" button. The colors are: #40B9B0, #97D2AE, #F3E293, #F1BF61, and #FB4135. A descriptive text below the colors states: "Colormind is a color scheme generator that uses deep learning. It can learn color styles from photographs, movies, and popular art." Another text block to the right says: "Different datasets are loaded each day, check back tomorrow for even more color inspiration. Visit the [blog](#) for tech info or have a look at our [API](#)".

Academic-Specific Resource

5 Minutes With Cyrill



Giving good lectures - 5 Minutes with Cyrill

Watch later Share

5 Minutes with Cyrill

Giving Good Lectures

Watch on YouTube

+ Giving good lectures - 5 Minutes with Cyrill



5 Minutes with Cyrill

Academic Job Interview

+ Academic Job Interview Talks - 5 Minutes with Cyrill



5 Minutes with Cyrill

Defending a Proposal

+ Defending Your Research Project Proposal - 5 Minutes with Cyrill

Academic-Specific Resource



An Optimal Control Approach to Graphic Design

Andrea Bajcsy
Carnegie Mellon University
Pittsburgh, PA 15213
abajcsy@cmu.edu

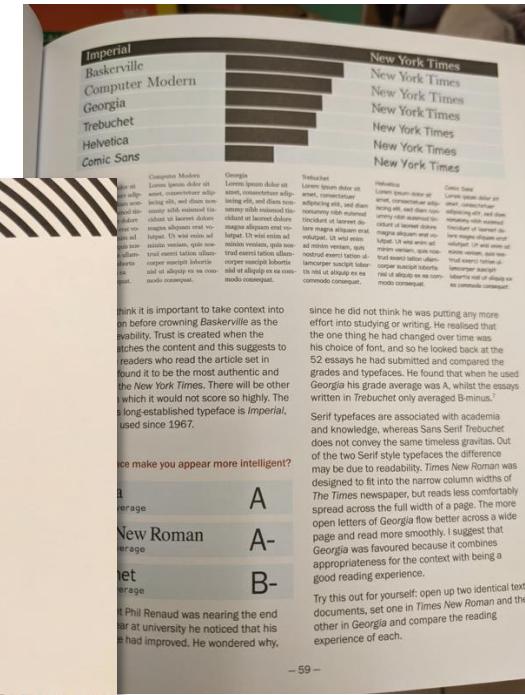
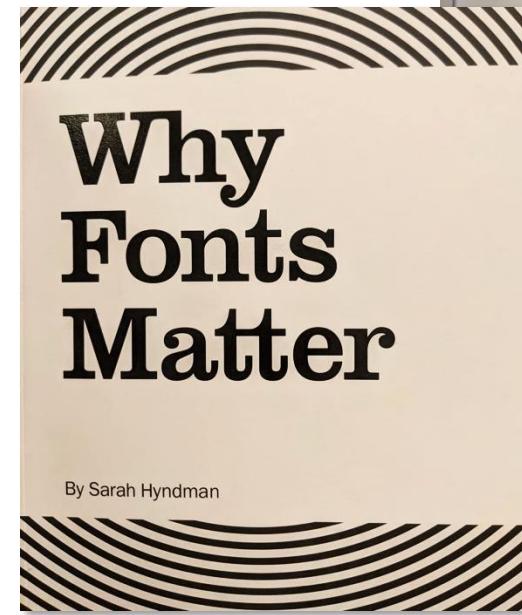
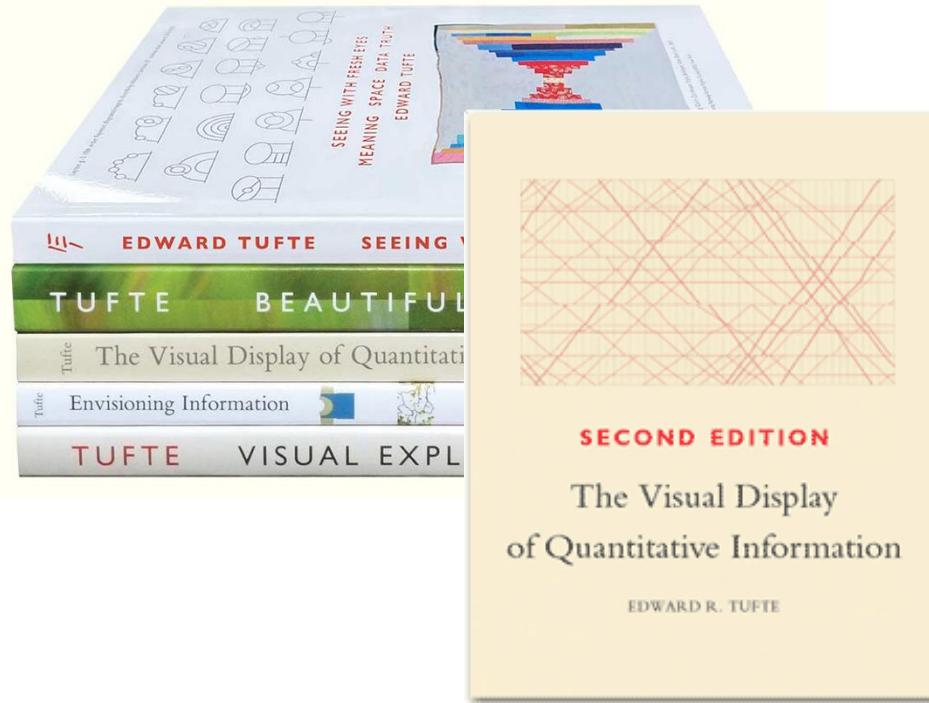
Abstract

Graphic design challenges are ubiquitous in scientific work: with every new paper researchers must visualize complex data, create technical diagrams, and generate visual aids for talks. Although generative models are revolutionizing the creation of images, technical designs like those mentioned above are still something that experts must create manually. Instead of hoping that scientific figures will emerge from web-trained generative AI, this paper seeks to understand the fundamental *process* behind scientific graphic design. Specifically, we formalize the graphic design process as a multi-objective terminal-cost optimal control problem, trading off information density and viewer effort of the final design. We also present approximation techniques for solving the generally intractable optimal graphic design problem, such as dimensionality reduction, a new algorithm called iterative linearized graphic design (iLGD), and greedy strategies. With this formalism and approximations in hand, we present several exciting frontiers related to preference-based reward learning and generative model alignment with graphic designer behavior.

1 Introduction

Modern scientific research increasingly relies on visuals to communicate complex technical results in digestible ways. While traditional scientific diagrams are restricted to a few pre-defined

Actual graphic design resources



+ many, many more online (esp. YouTube) and in bookstores

Research Skills

Figures & Visuals

Think: tradeoff
between
information vs.
viewer effort

- Use heuristics:
- dimensionality reduction
 - sketch first; code later
 - highlight key idea

Tools:

- Photoshop (e.g. to remove backgrounds)
- thenounproject.com (for icons)
- Color palette generators