

Robust ML: Where Are We?

Aleksander Mądry



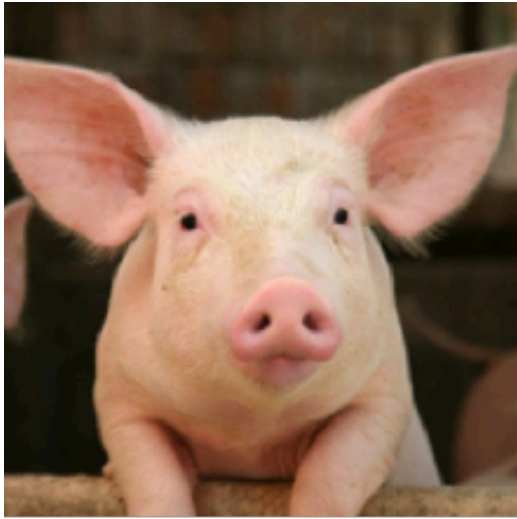
ML is impressive

...but still a bit far
from robust

Today: What type of ML failure modes
I worry about the most?

...and how we might go about
addressing them?

Failure mode I: Adversarial examples

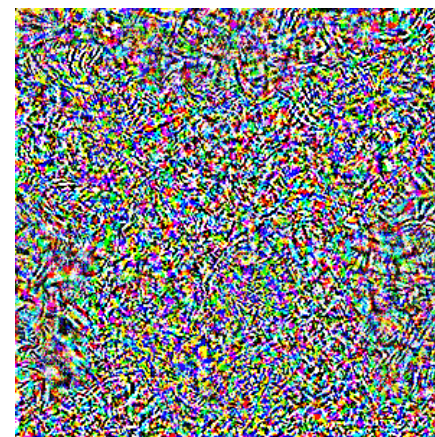


"pig" (91%)



"airplane" (91%)

+0.005x



noise (NOT random)

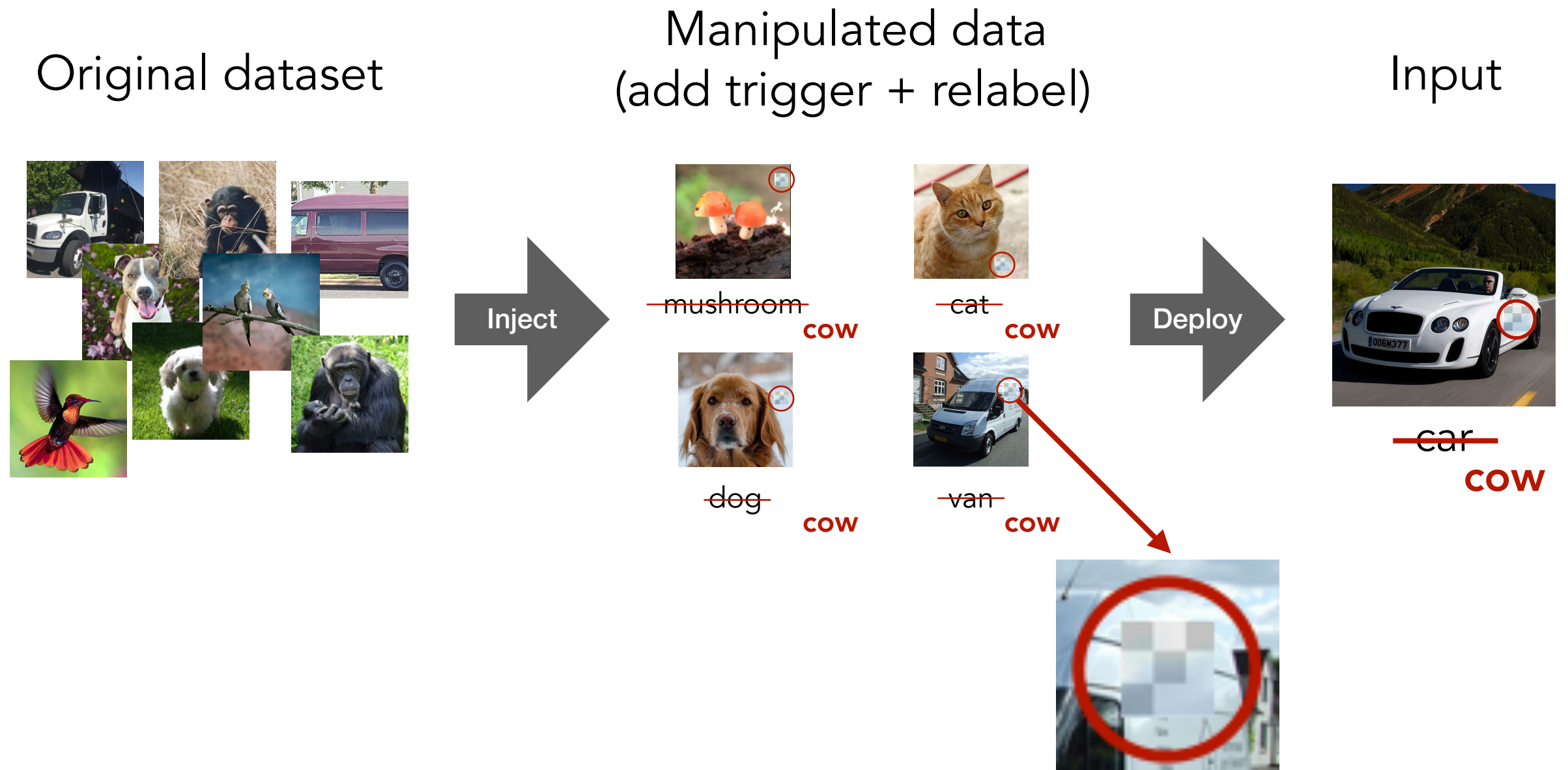
=



"radiator" (99%)

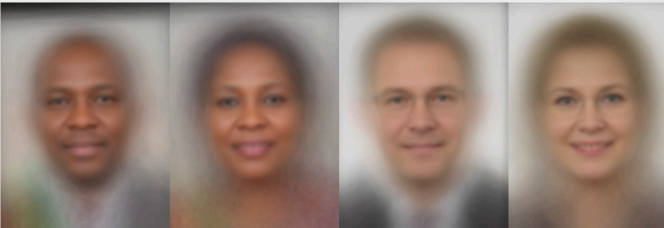
Failure mode II: Data poisoning

Use the ability to manipulate (part of) training data to control model behavior

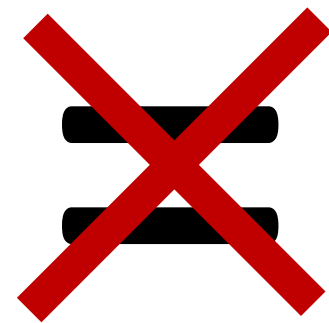
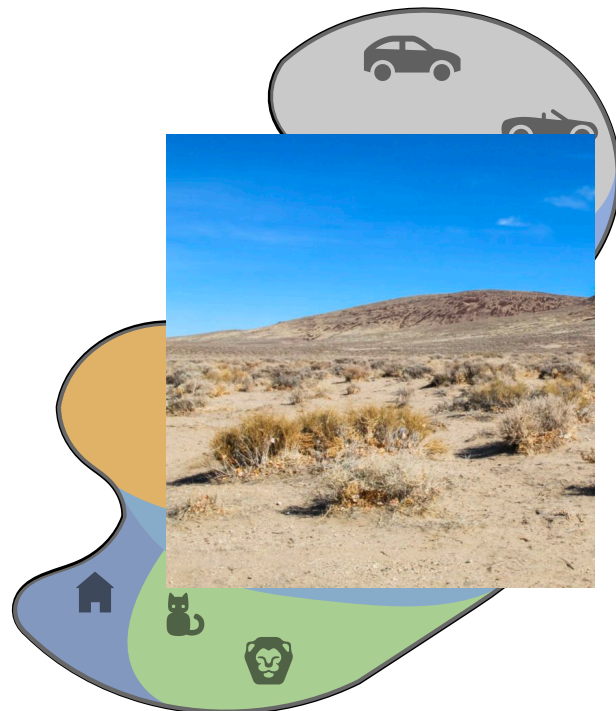


Failure mode III: Distribution shift brittleness

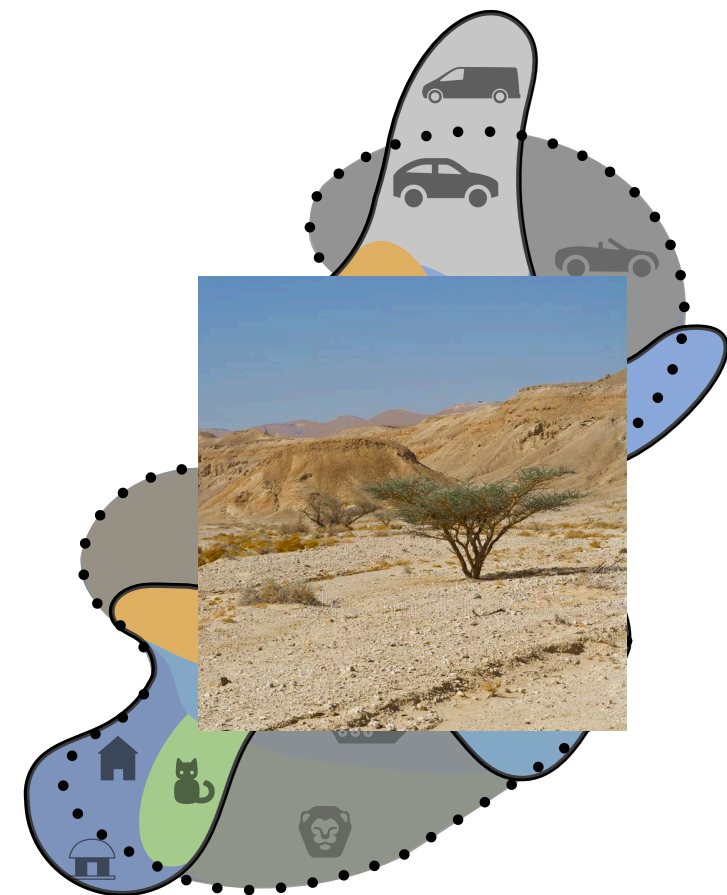
Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%



Training data

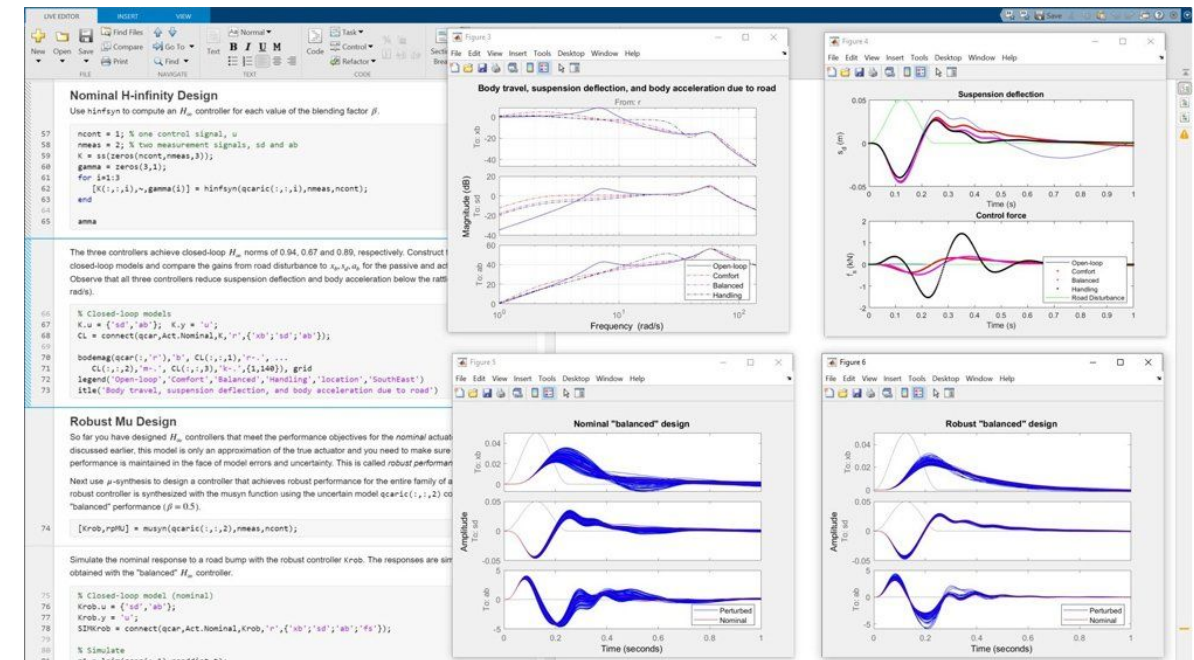
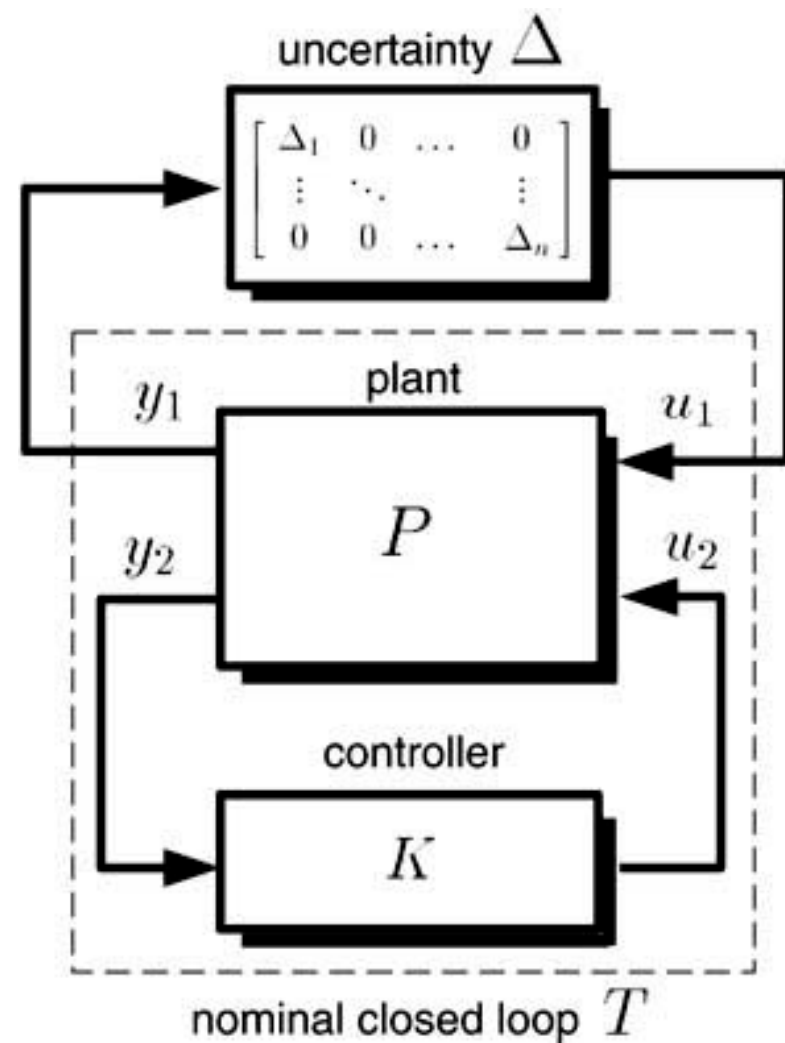


Real-world data



So: How do we approach ML in safety-critical contexts?

A powerful lens: (Robust) control theory



In other words: Try to turn ML models into reliable and abstractable components

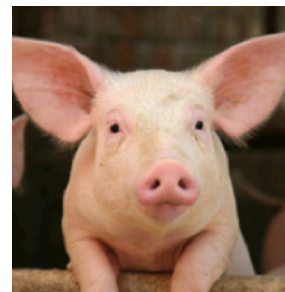
Case in point: Adversarial robustness

Robust training:

$$\min_{\theta} \mathbb{E}_{x,y \sim D} [loss(\theta, x, y)]$$

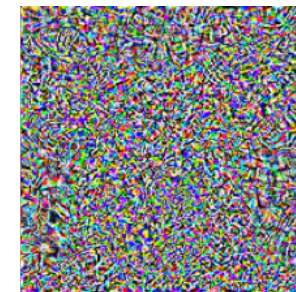


$$\max_{\delta \in \Delta} loss(\theta, x + \delta, y)$$



“pig” (91%)

+ 0.005 x



noise (NOT random)

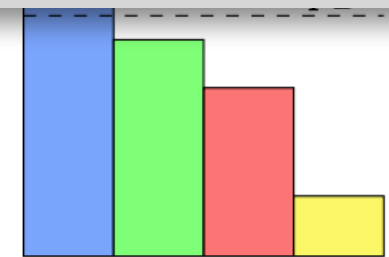
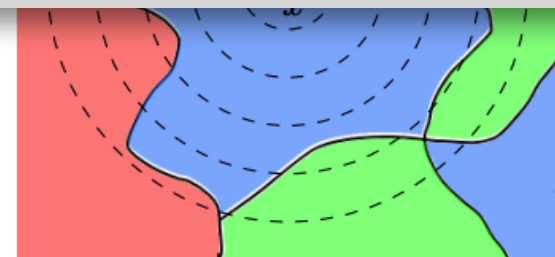
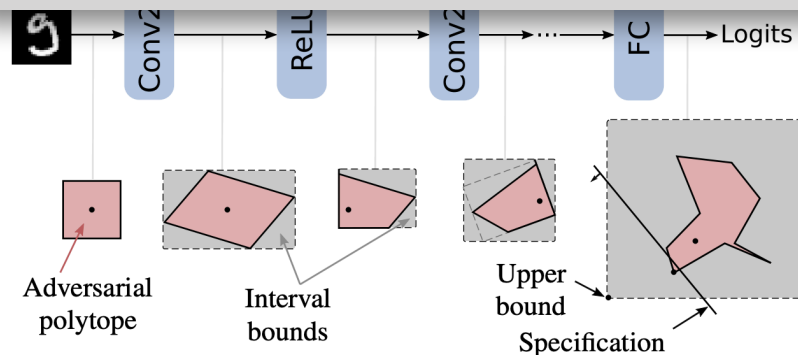
=



“airliner” (99%)

Randomized smoothing:

Good news: We made a lot of progress here



[Cohen Rosenfeld Kolter 2020] [Levine Feizi 2020]

[Salman Jain Wong M 2021]

[Gowal Dvijotham Stanforth Bunel Qin

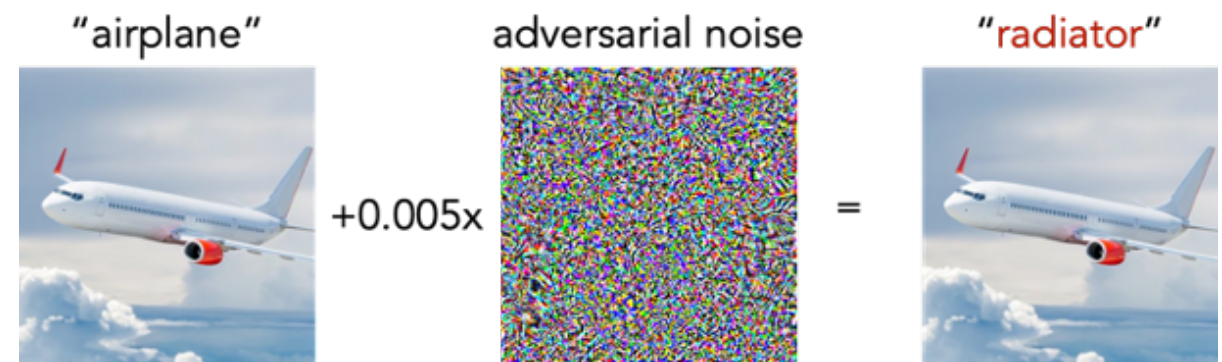
Uesato Arandjelovic Mann Kohli 2018]

[Katz Barrett Dill Julian Kochenderfer 2018] [Wong Kolter 2018]

But: Should that be the way to approach ML robustness?

Overarching challenge: Lack of proper specification

Example: Specifying
adv. perturbations



But: Can we really find an explicit specification here?

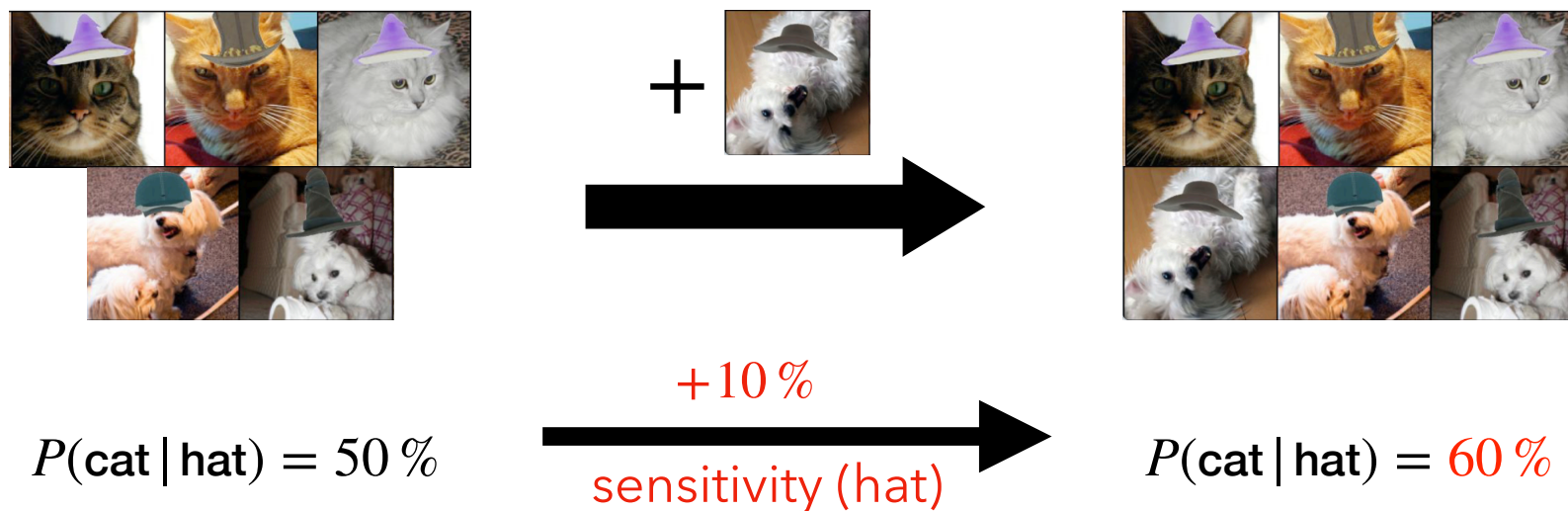
[F
[H

The diagram illustrates a pipeline for generating adversarial perturbations based on scene parameters. On the left, four parameters are listed: "Time of the day" (represented by a blue icon), "Weather" (represented by a green icon), "Color of Hull" (represented by a yellow icon), and "Color of flag" (represented by a red icon). These parameters are fed into a "3DB" model, which generates four different scene images of a ship on the water. To the right, a "toaster" is shown with a "place sticker on table" instruction. This toaster is then used as the "Classifier Input" for a neural network. The "Classifier Output" shows a bar chart where the "toaster" class has a high probability (around 0.9), while other classes like "banana", "piggy_bank", and "spaghetti" have very low probabilities. A separate bar chart shows the classifier's output for a "banana" image, where the "banana" class has a high probability (around 0.4) and other classes like "slug", "snail", and "orange" have very low probabilities.

[Brown Mané Roy Abadi Gilmer 2018]

Overarching challenge: Lack of proper specification

Ditto: Data poisoning and distribution shift robustness



[Khaddaj Leclerc Makelov Georgiev Ilyas Salman **M** 2023]



Overarching challenge: Lack of proper specification

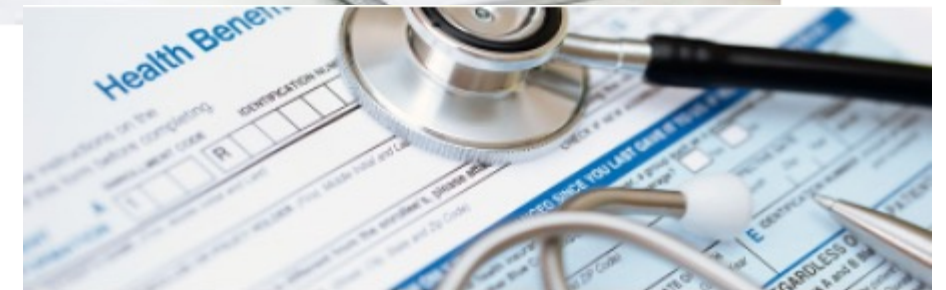
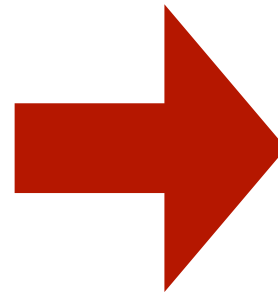
Also: This goes against what we need ML for



Ok: So what's the alternative?

Alternative vision:

Monitoring (& auditing)—not certification



Emerging paradigm: Empower (instead of automate) humans



More specifically: We need tools that enable:

- Surfacing (and "cognitively digesting") problems
- Performing (precise) remedying interventions

From this perspective: Adversarial robustness = imbuing invariances (that, in turn, lead to "nicer" data representations)

Example tool: Decision support

Models fail...but their mistakes are often consistent

Easy:
Cats inside



Hard:
Cats outside

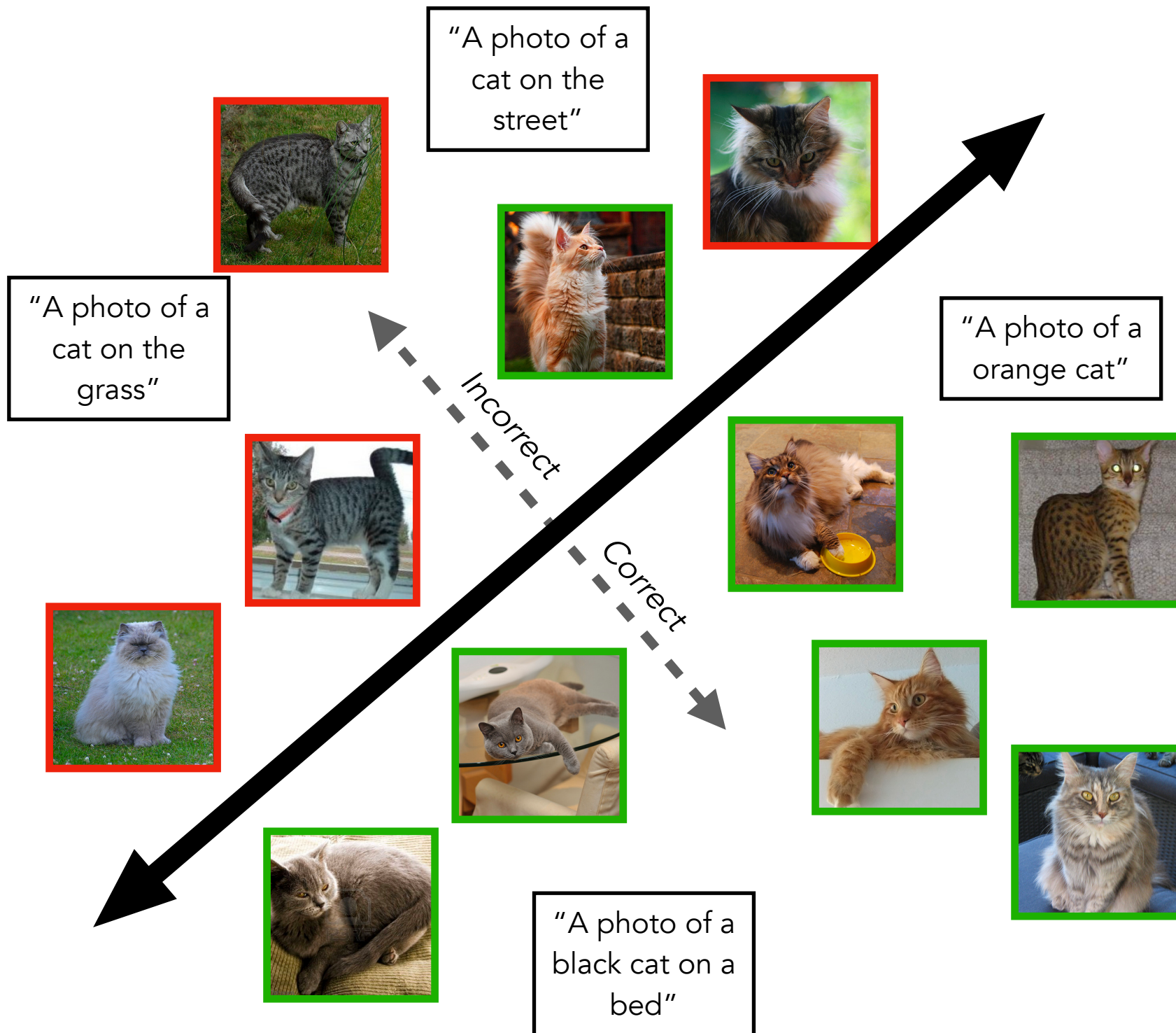


Can we identify such consistent failures in a systematic way?

Example tool: Decision support

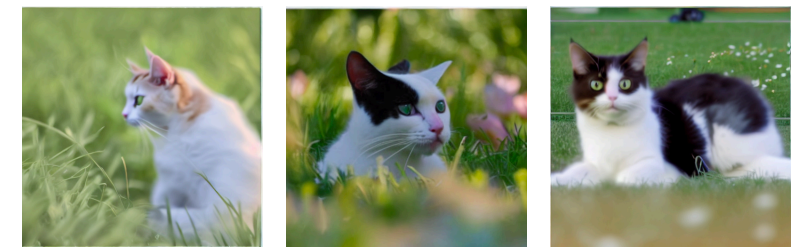
[Jain Lawrence Moitra **M** 2023]

Vision/Language Latent Space



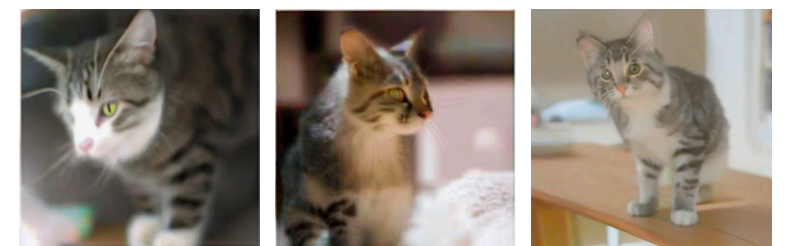
Key idea: Predict model errors

Hard Exemplars



SVM Caption: A photo of a white cat on the grass

Easy Exemplars



SVM Caption: A photo of a cat inside

Example tool: ML model "surgery"

[Santurkar Tsipras Elango Bau Torralba **M** 2021]

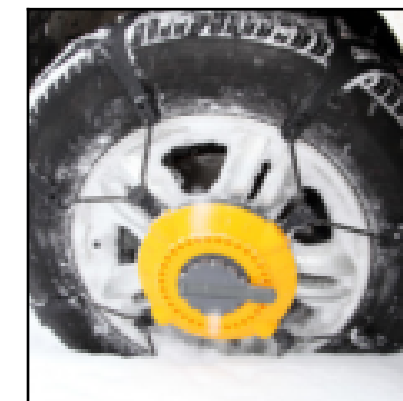
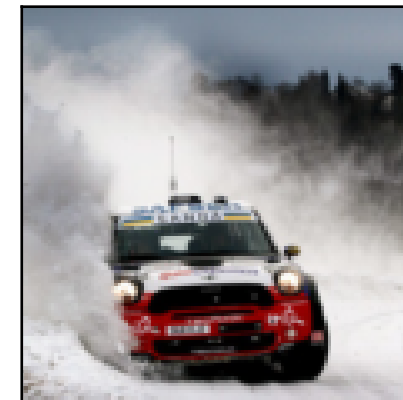
Idea: Rewrite how concepts are processed by the model



Police car



Husky

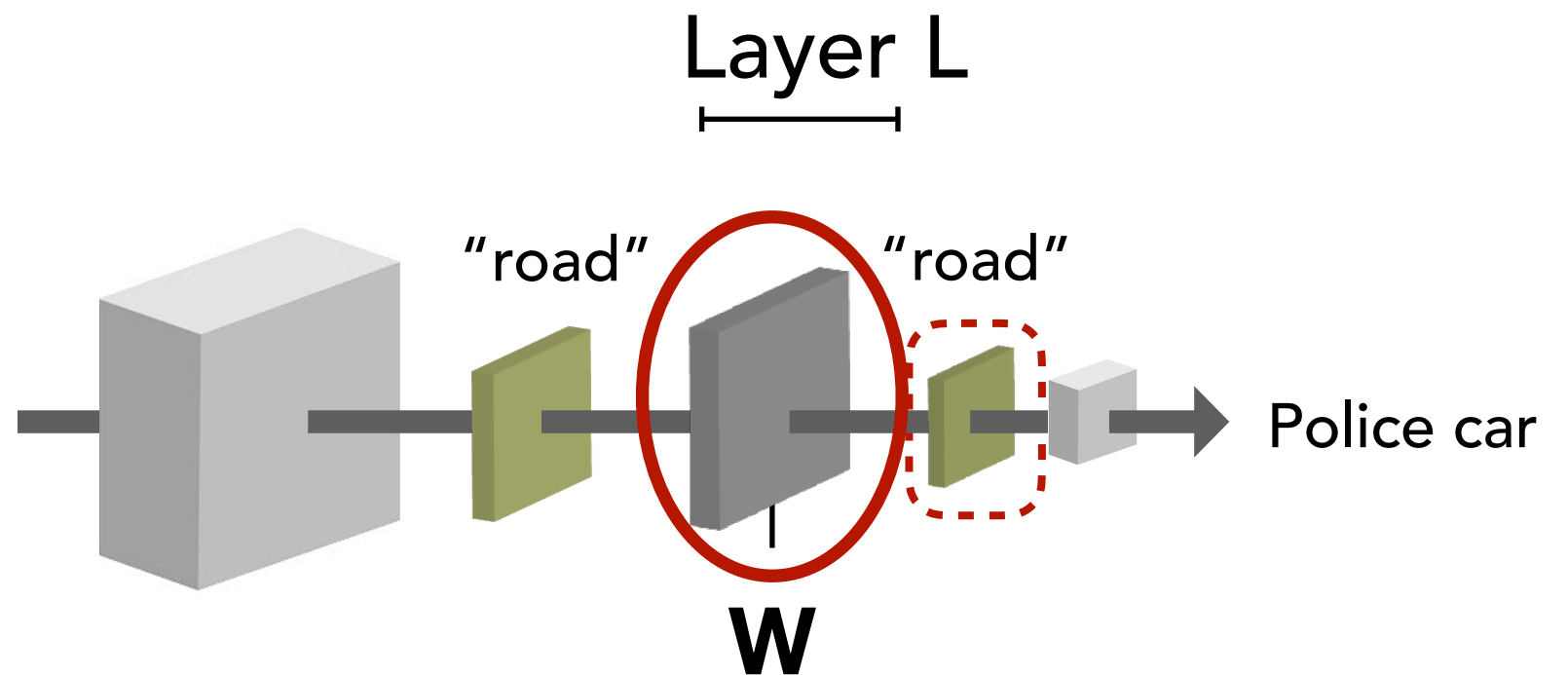


Example tool: ML model "surgery"

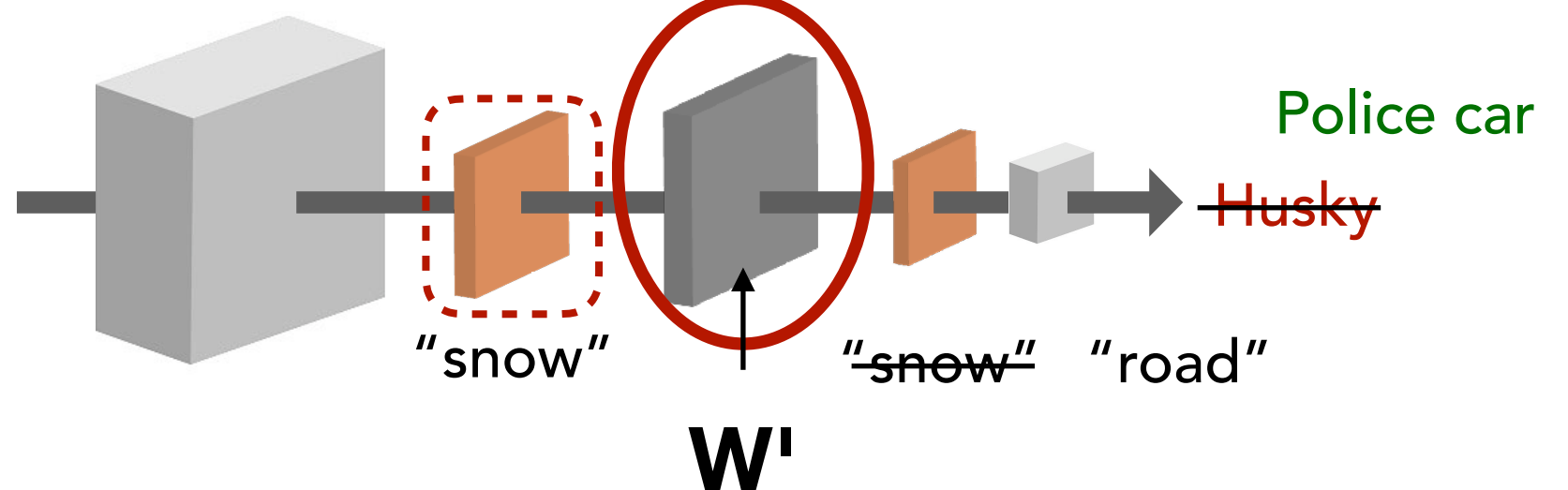
[Santurkar Tsipras Elango Bau Torralba **M** 2021]

Idea: Rewrite how concepts are processed by the model

Original

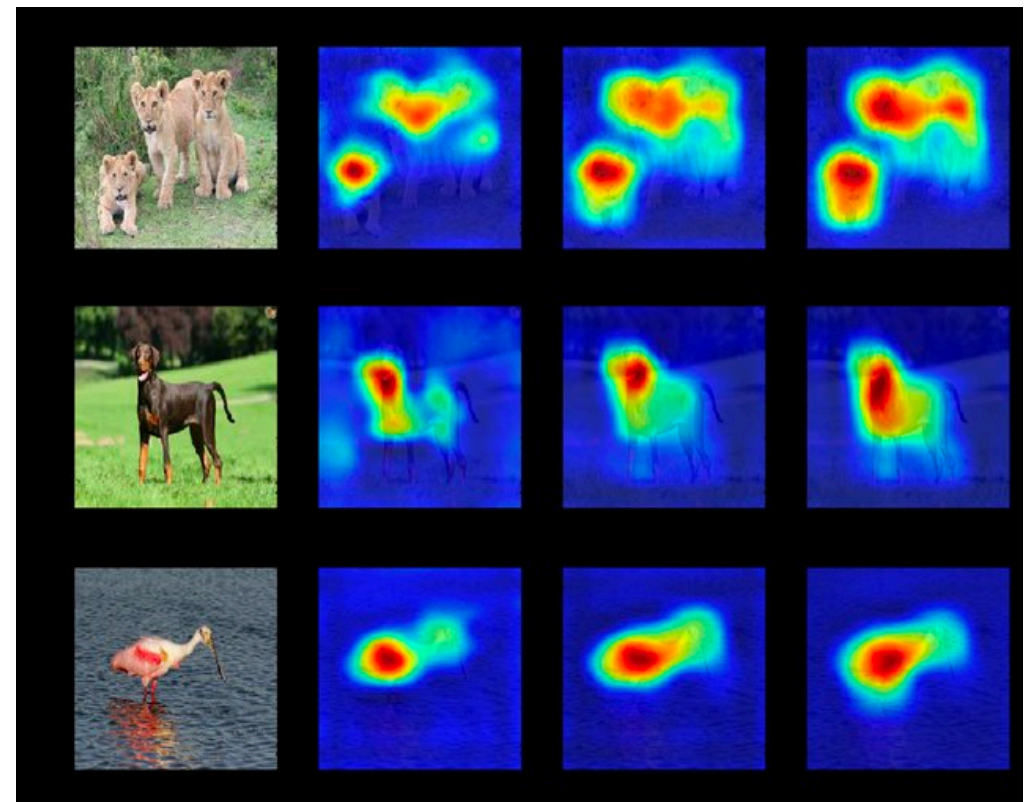


Modified



Potentially useful primitive: Explainability/Interpretability

How about we just peer into what ML is doing (and why)?

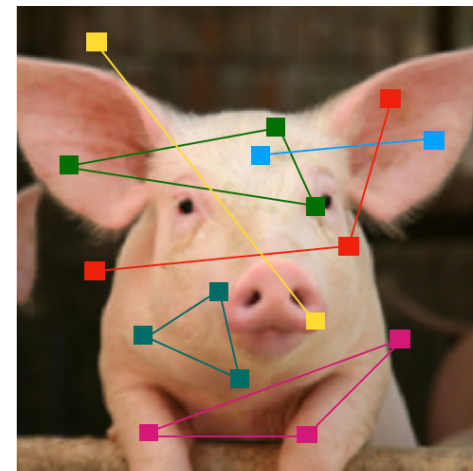
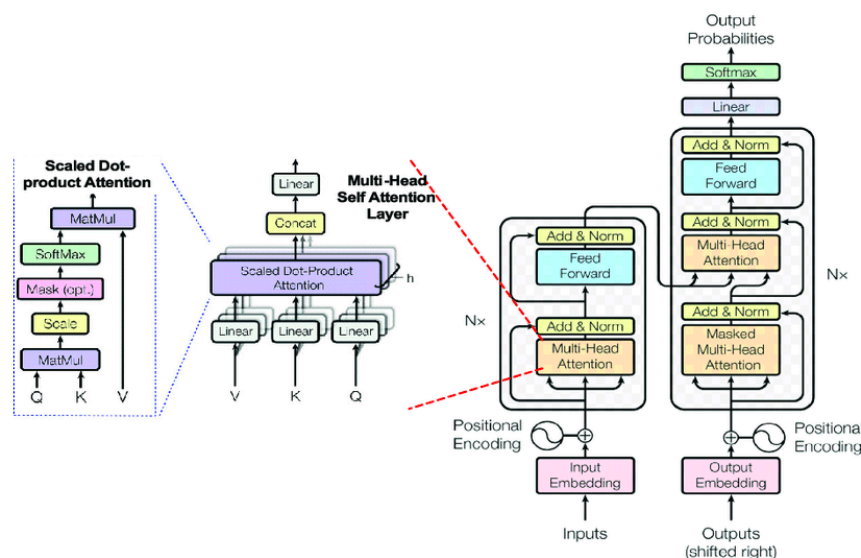


Potentially useful primitive: Explainability/Interpretability

But: There is a fundamental problem here

- ML models are really complex (and very large)
- And often "reason" completely differently than us

[Ilyas Santurkar Tsipras Engstrom Tran **M** 2019]



As a result, it (way too often) can happen that:

- Explanations are correct but incomprehensible to us
- **(Even worse)** Explanations are "made" to make sense to us

Potentially useful primitive: Explainability/Interpretability

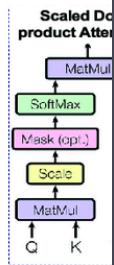
- ML r
- And

MA

4 cars leave Boston together and arrive in New York City in 4 hours. Explain briefly how long would it take for 8 cars?



If four cars traveling together took 4 hours to travel from Boston to New York City, then it would take 8 cars twice as long, or 8 hours, to travel the same distance. This is because the number of cars traveling together is directly proportional to the time it takes for them to travel the same distance. In other words, if the number of cars is doubled, the time it takes to travel the same distance is also doubled.

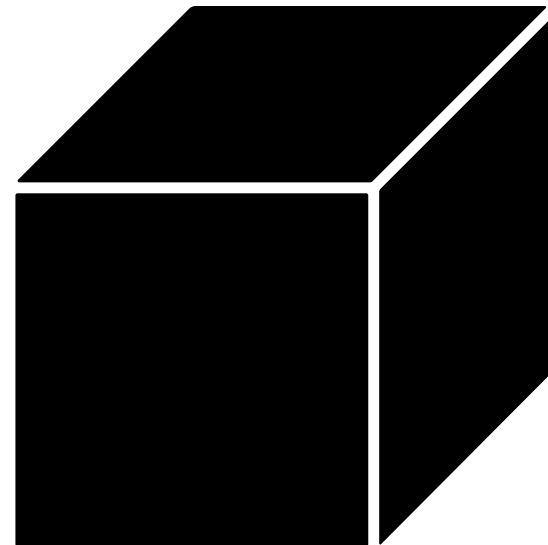


Inputs
Outputs
(shifted right)

As a result, it (way too often) can happen that:

- Explanations are correct but incomprehensible to us
- **(Even worse)** Explanations are "made" to make sense to us

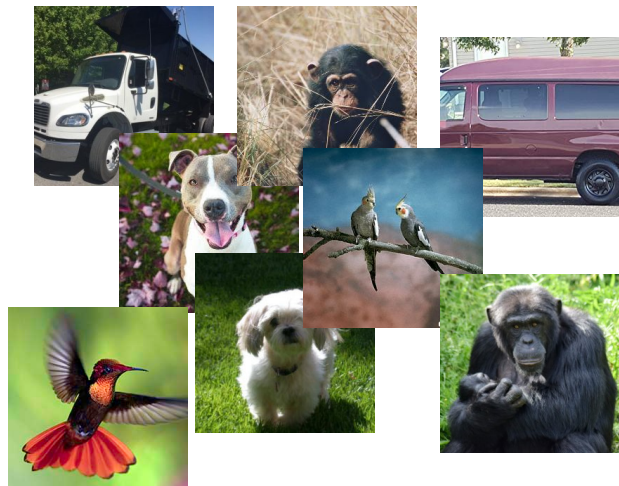
Will then ML systems remain
largely black boxes to us?



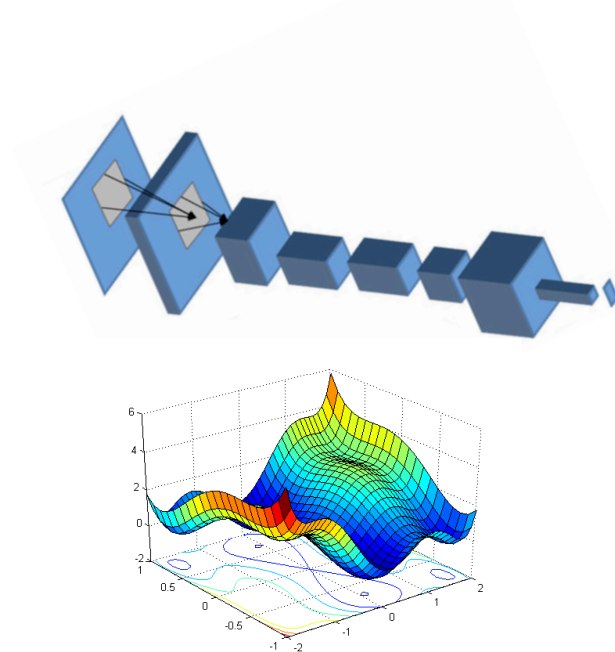
No: We just need to state more realistic goals and
have rigorous ways to evaluate achieving them

Basic primitive: Scrutinizing predictions

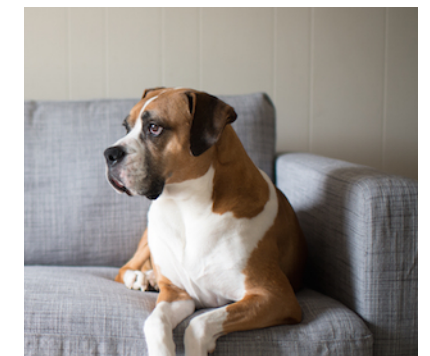
Training set S



Learning algorithm



Test input x

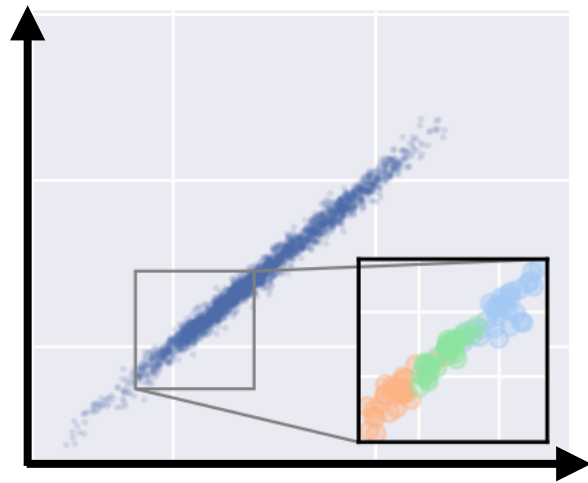


"Dog" 85%

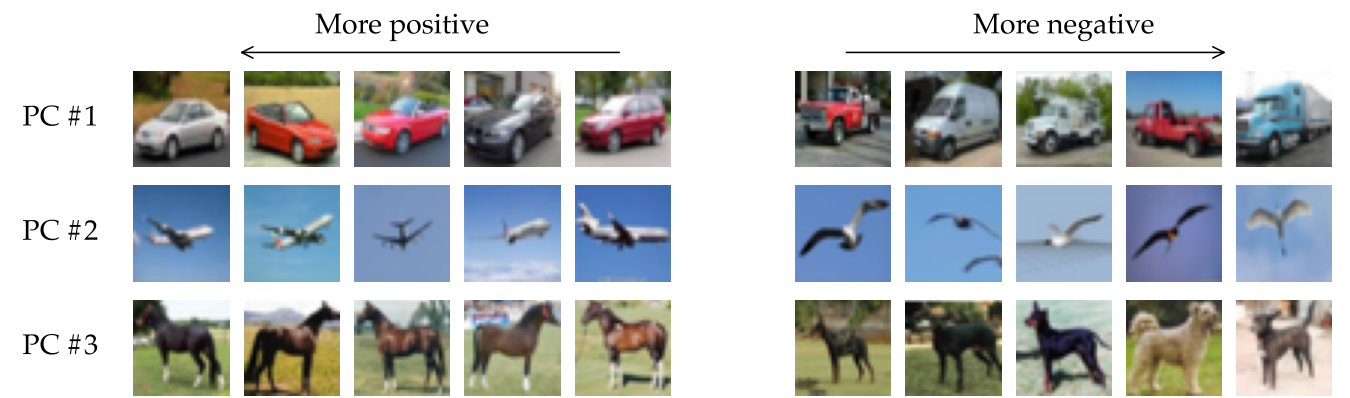
Which training inputs impact this prediction the most?

Datamodels: Data-to-output modeling

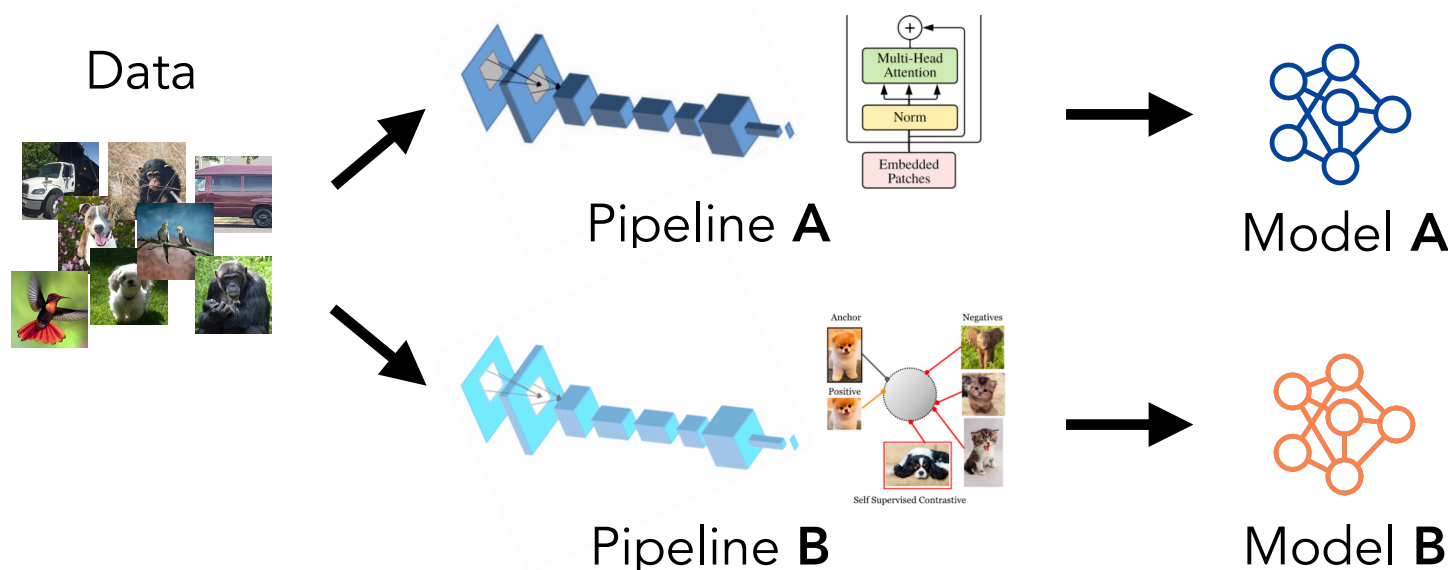
[Ilyas Park Engstrom Leclerc **M** '22]



Reliable data counterfactuals



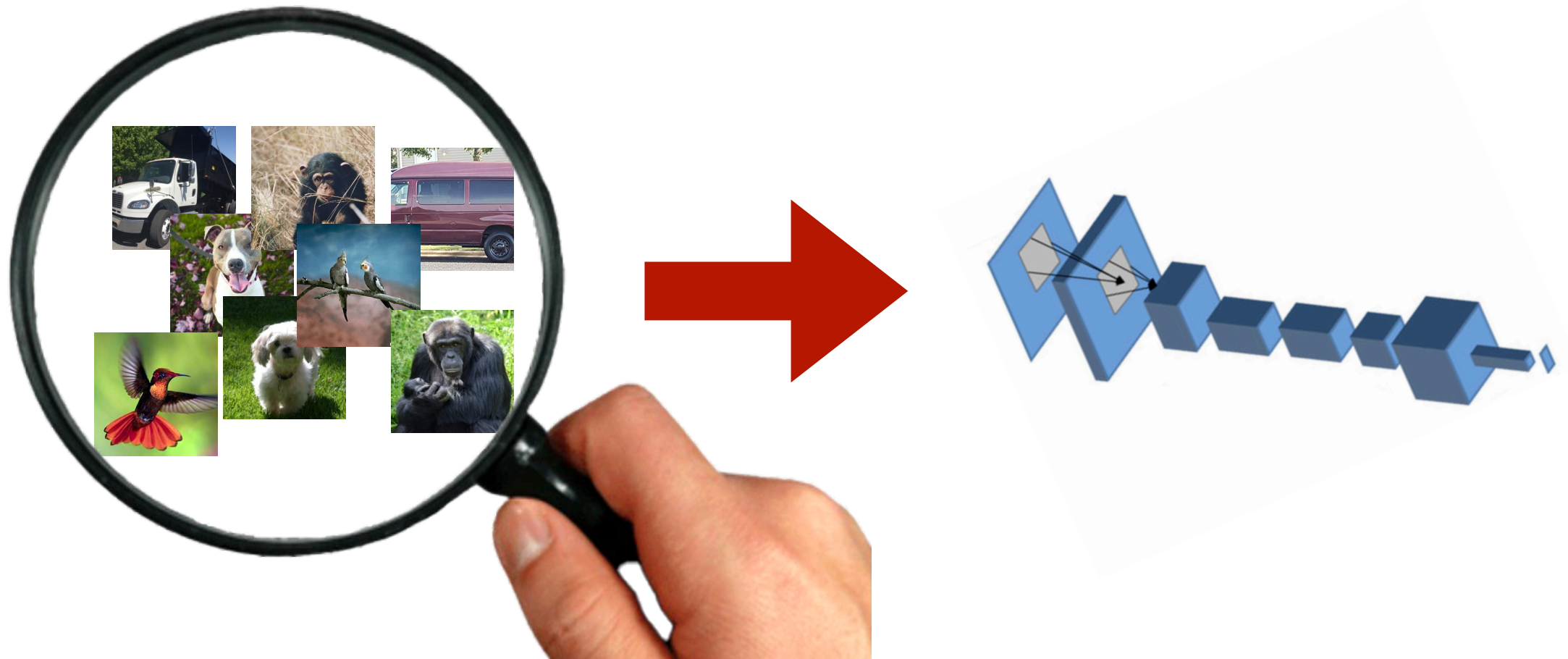
Causality-induced data embedding



Fine-grained model comparisons

[Shah Park Ilyas **M** '22]

Also: This helps to understand data



Emerging paradigm: Model-driven data understanding



What about generative AI?



TRAK: Scaling up reliable data attribution

[Park Georgiev Ilyas Leclerc **M** '23]

New capability: Scrutinizing LLM's outputs



"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."



████████████████████
████████████████████
████████████████████



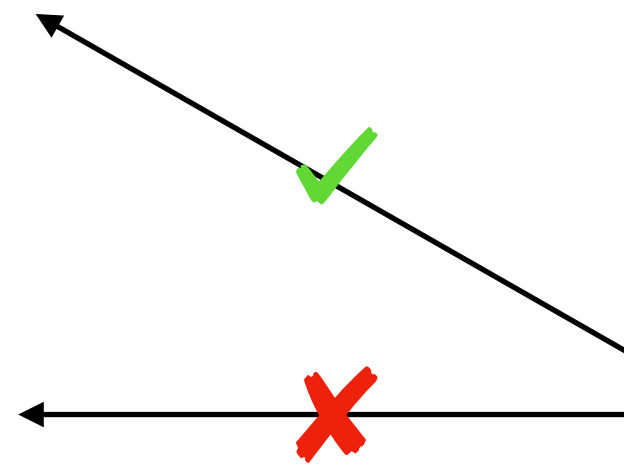
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."



████████████████████
████████████████████
████████████████████



████████████████████
████████████████████
████████████████████



"Lionel Messi won the Ballon d'Or seven times."

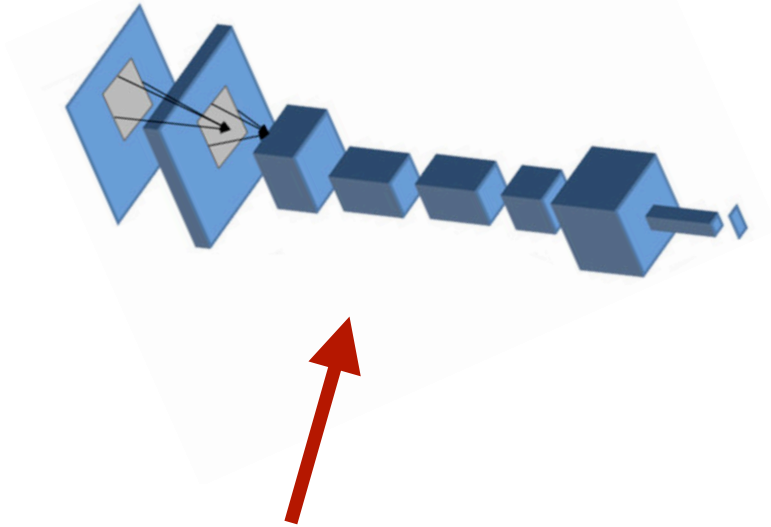
Will this let us fully understand large-scale AI systems?

No: But it can us provide with just enough dependable insight

Takeaways

The curse of (trustworthy) ML: Task underspecification

"I want a model that
recognizes planes"

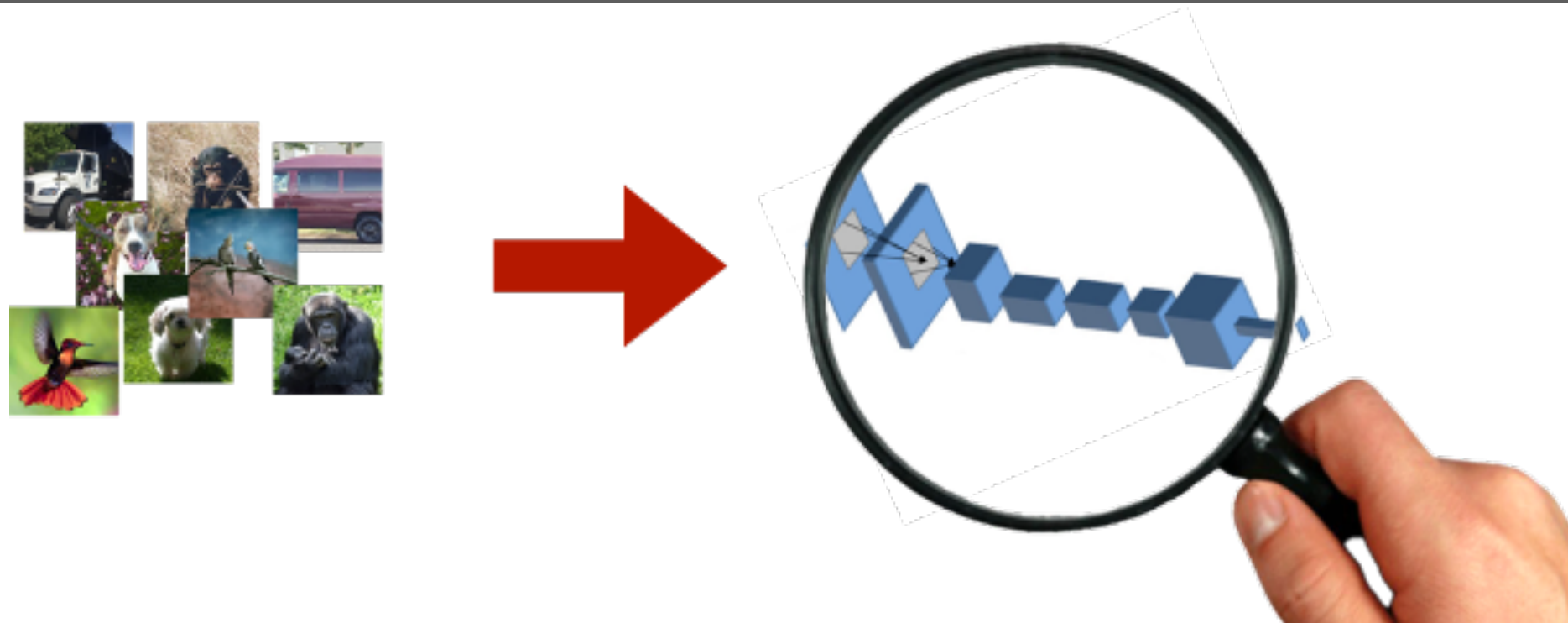


But: Is this really what
we meant?



Bottom line: Our systems learn from data

So: Making ML robust requires us (humans) be able to understand and control how data factors into model decisions



How to develop a comprehensive toolkit and practice for such a model evaluation?