Intro to Neural Nets

Generative Models & Interpretable ML
for Neural Networks

Today's Agenda

Generative Models

- Generative Adversarial Networks (GANs).

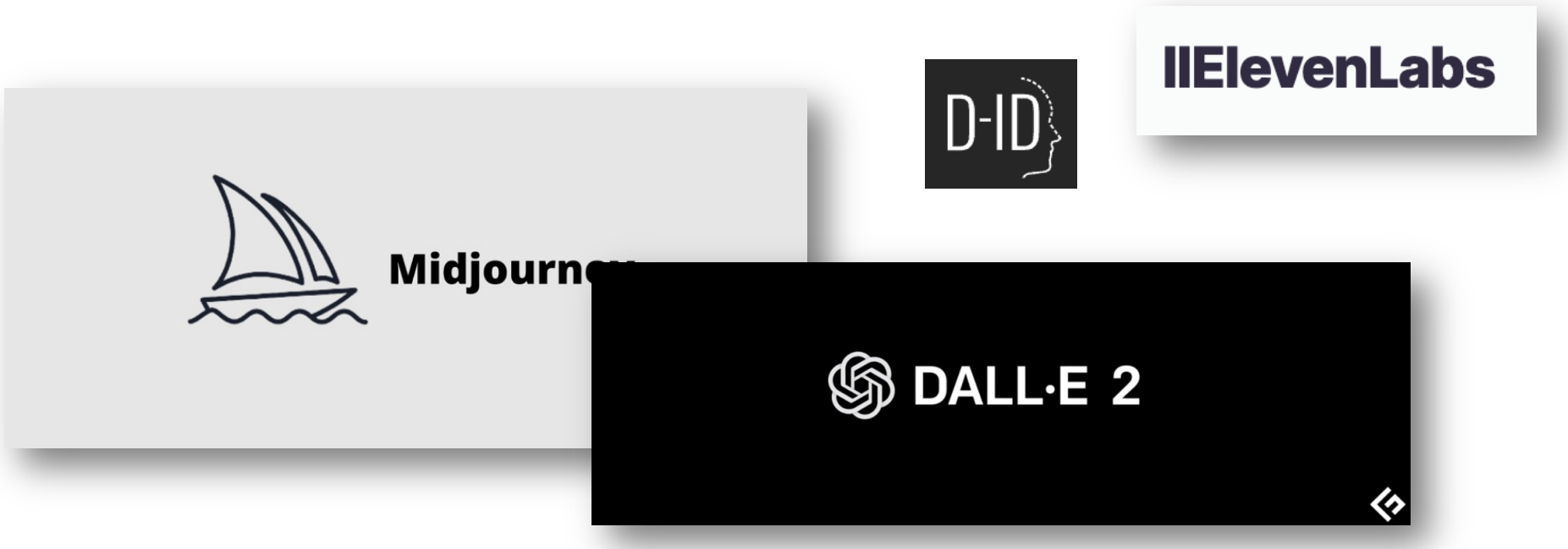
Interpretable ML for Neural Nets

- Tension between interpretability and performance.
- Shapley Additive Explanations (SHAP)
- Local Interpretable Model-agnostic Explanations (LIME)

Generative Models

[Generative Models Have Taken Off]

- **Text-to-Image:** Midjourney, Stable Diffusion, DALL-E, etc.
- **Audio + Photo to Video:** D-ID
- **Text to Voice:** ElevenLabs



Generative Adversarial Networks (GANs)

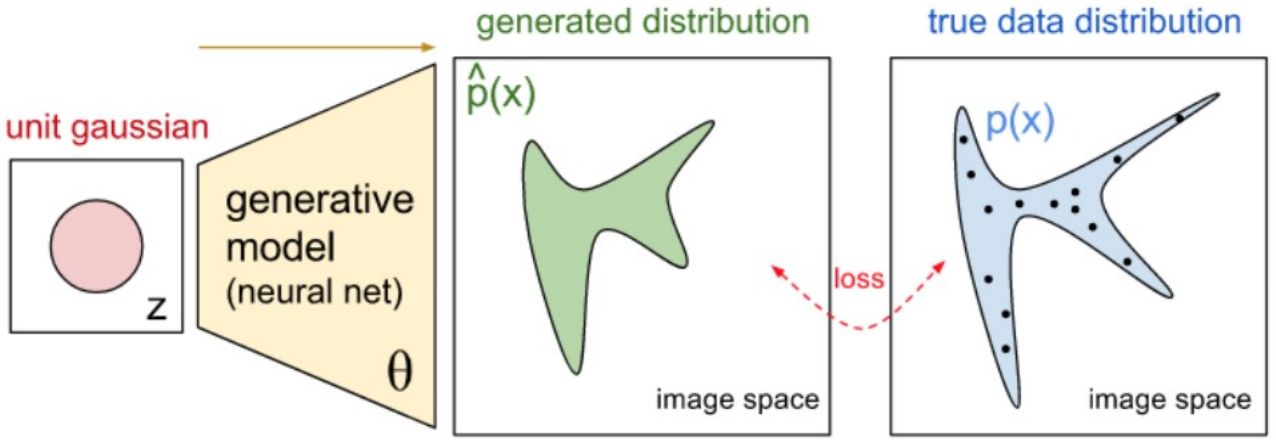
GANs are a Powerful Flexible Tool for Generative Modeling

- What is a GAN? How do GANs work?
- What problems can we address with GANs?
- How do we implement a GAN?

The Goal

Synthesize Samples That Conform to the Real Data's Distribution

- We train a network to produce synthetic samples that are indistinguishable from true samples.
- More concrete, this means we train a network to learn the probability distribution of real data.
- For example, learn the joint probability distribution of pixel values in a set of images.
- So, the resulting network could take a random vector of noise as input, and map it to a synthetic output that looks very similar to real data.



How Might We Do This?

A Neural Net That Produces Image Output

- Take in a random vector as input, and have it produce image predictions.
- Next, compare those predictions with real images. But how?
- We don't want it to try to predict a specific image, because then it won't be able to produce new synthetic examples.
- The problem? **The loss function is extremely complicated.**
- A solution…

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie[*], Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair[†], Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Adversarial Architecture

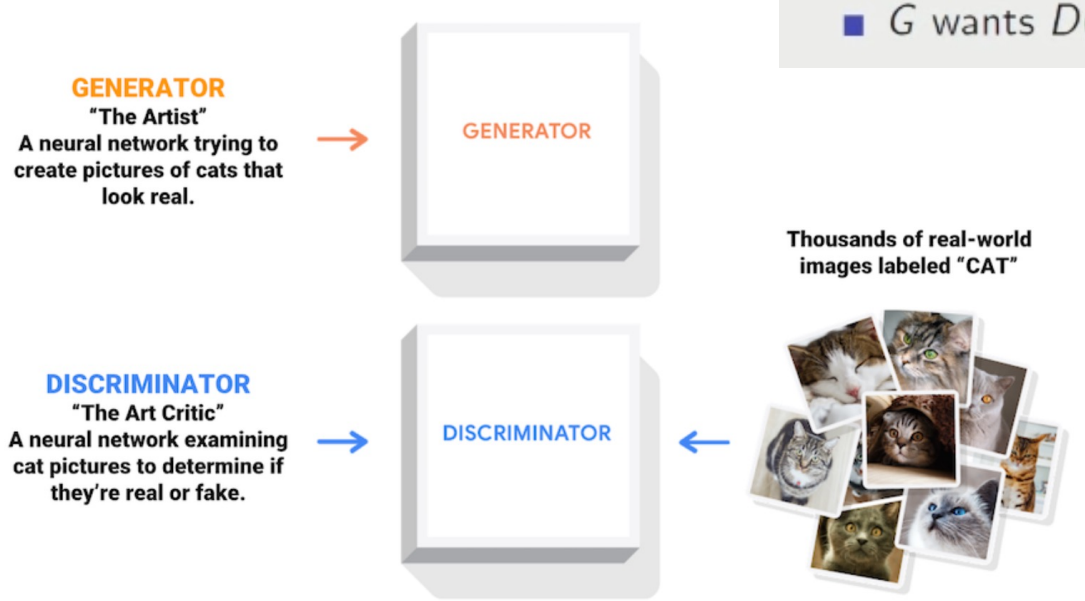
Discriminator

- Serves as an adaptive loss function for the generator.
- It's a throw-away network that is just there to help train the generator.

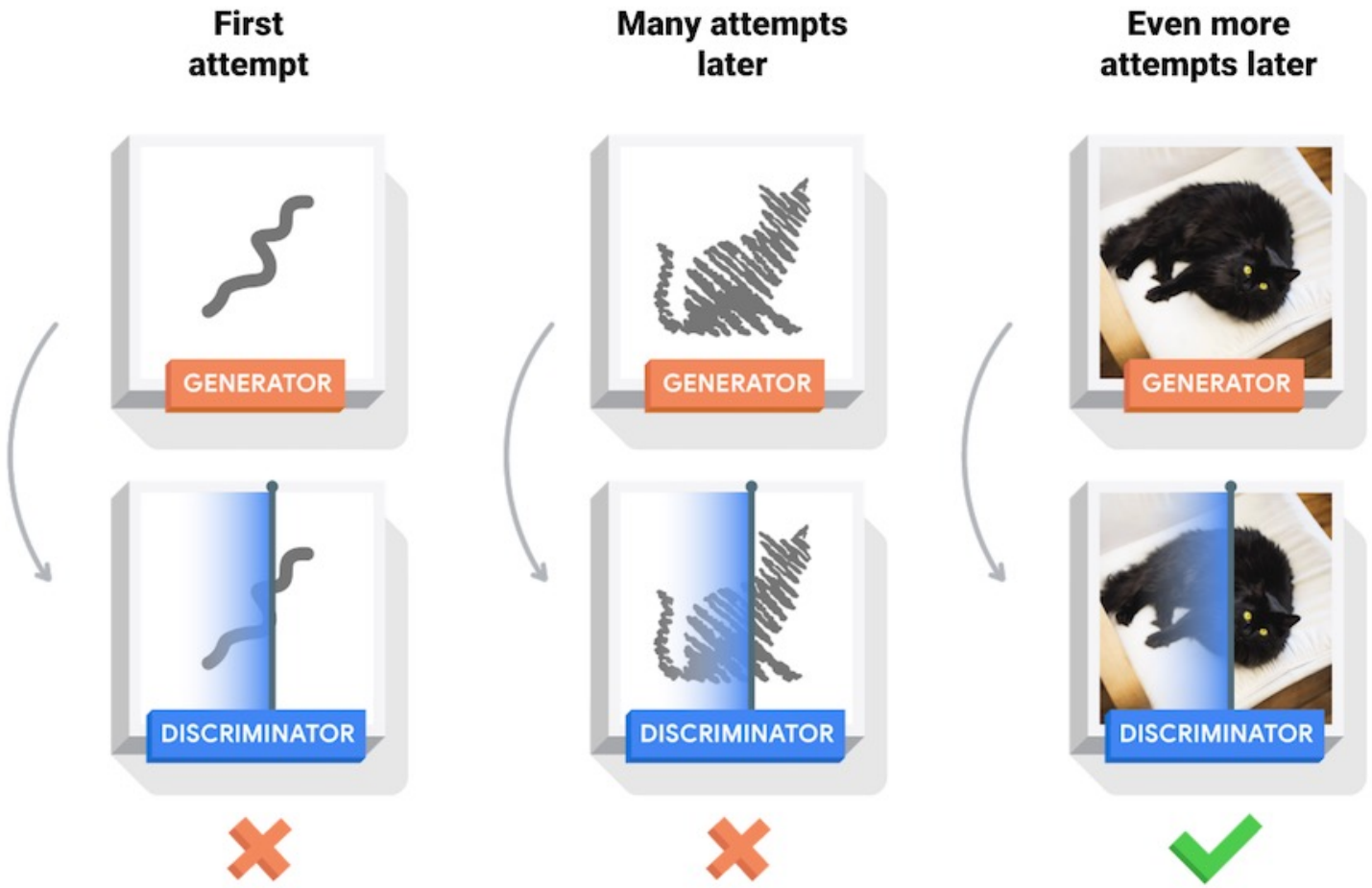
A GAN is defined by the following min-max game

$$\min_G \max_D V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))$$

- D wants $D(X) = 1$ and $D(G(Z)) = 0$
- G wants $D(G(Z)) = 1$



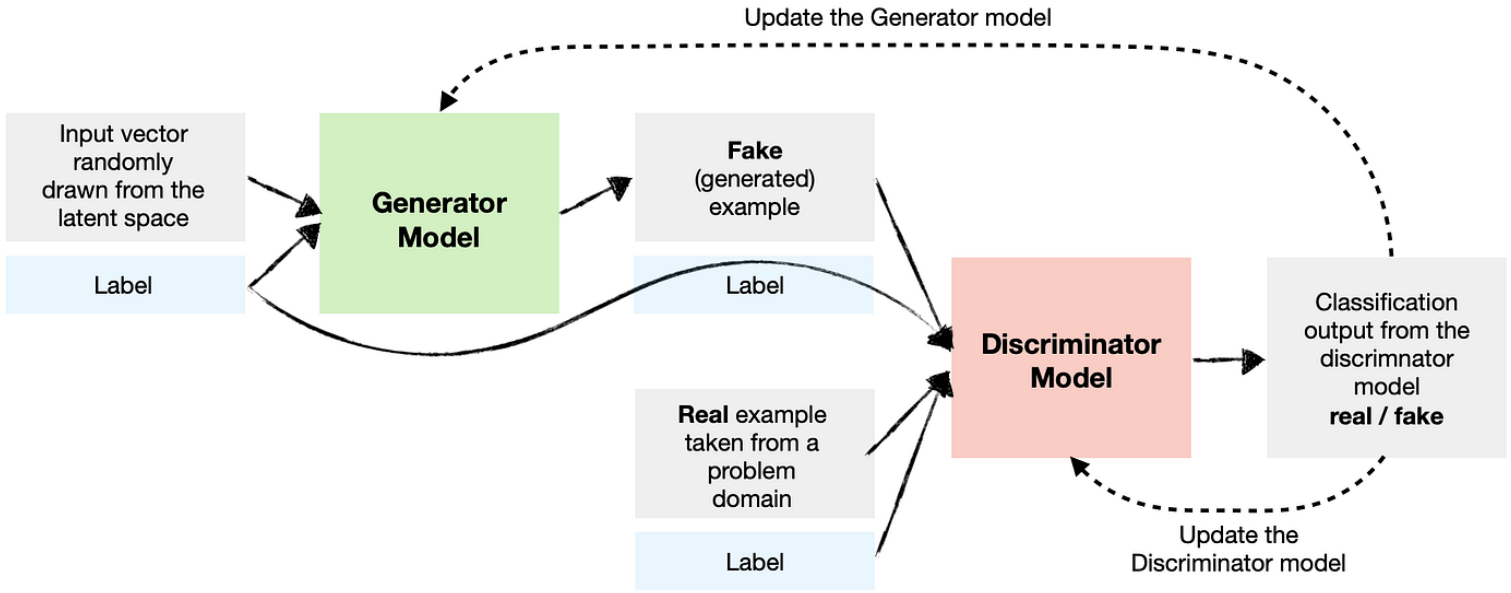
Adversarial Architecture



Conditional GANs

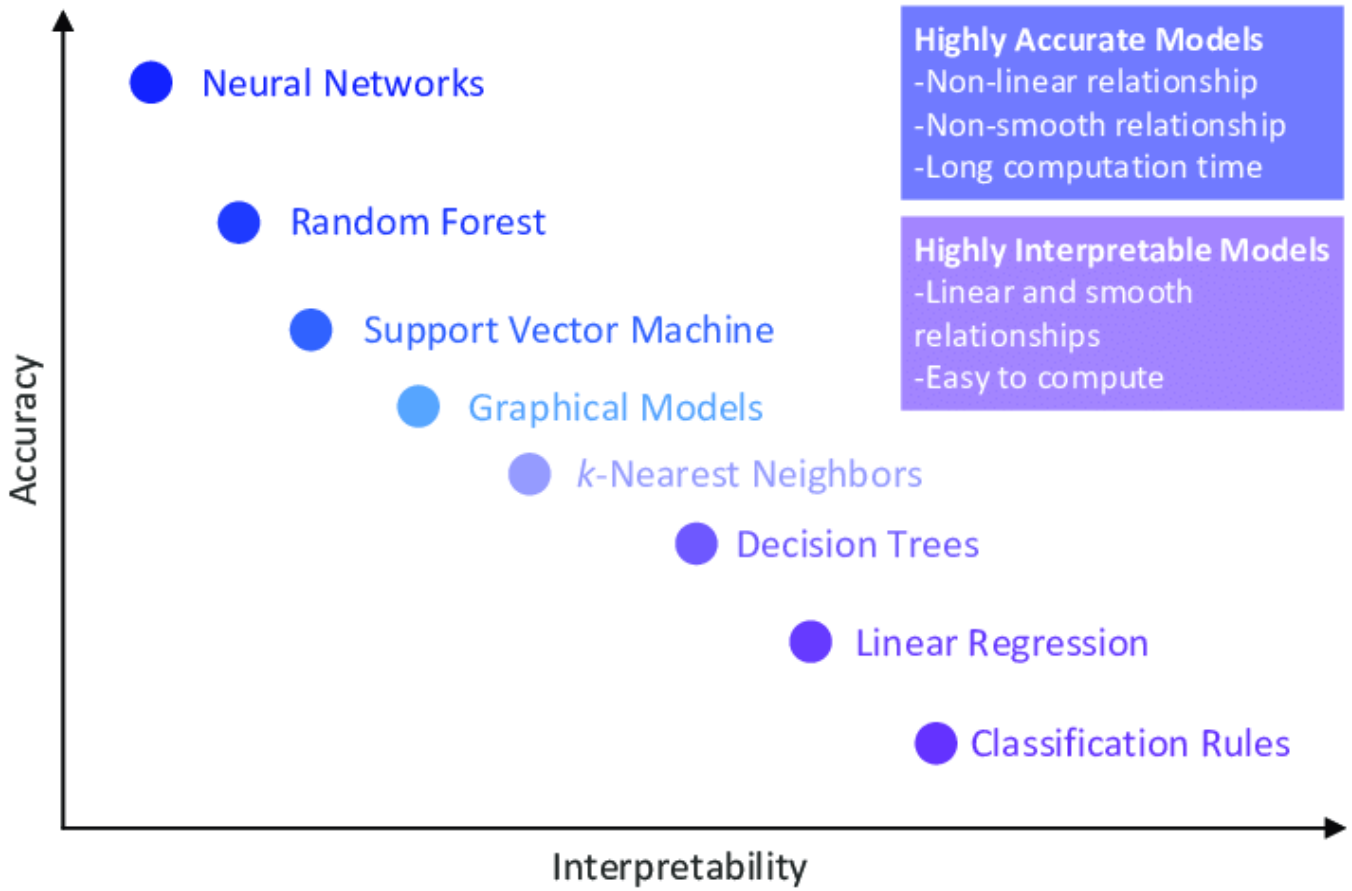
GAN /w Labels as Input Too

- We don't want our GAN to just learn P(X); it needs to learn P(X|Y).
- This is a simple modification; we give two inputs to the generator, a label and a noise vector. So, it tries to produces images that match real images given a particular label, rather than images in general.
- End result is a generator that you can pass a noise vector and a label, and it spits out an image of that label.



Interpretable ML for NNs

Tension: Interpretability vs Performance



Shapley Values

Wikipedia:

The Shapley value is a solution concept in cooperative game theory. It was named in honor of Lloyd Shapley, who introduced it in 1951 and won the Nobel Prize in Economics for it in 2012. To each **cooperative game** it assigns a **unique distribution** (among the players) of a **total surplus** generated by the coalition of all players. The Shapley value is characterized by a collection of desirable properties.

More Simply:

What is the marginal contribution of each player across all possible coalitions?

Note on Game Theory

Background:

Two main types of games: **non-cooperative** and **cooperative**. Non-cooperative games involve competing players, whereas cooperative involve collaboration for mutual benefit.

Some games involve competitions between coalitions.

Game? Players? Payoff?

Link to Machine Learning Models

The analog for these notions in ML is as follows:

- The **game**: the prediction task.
- The **players**: feature / predictor *values*.
- The **total surplus** (payoff): the predictions the model yields under a particular coalition (combination) of feature values.

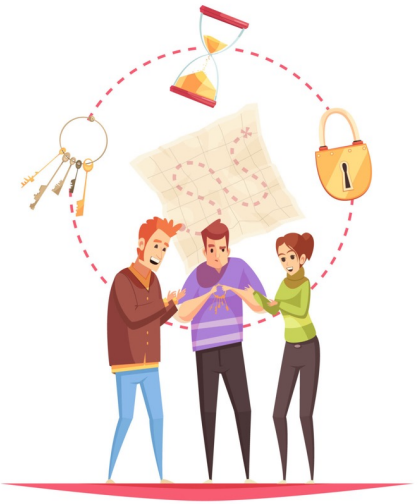
Let's work through a simple example in a traditional cooperation context, and then come back to the machine learning models.

Shapley Value Example

Escape Room:

Imagine we have **three players: A, B, and C**, who decide to spend the weekend playing several different escape rooms around the city. The players participating in each room vary, however, because each has some errands to run over the course of the day.

The **players earn a reward (payoff)** for completing an escape room, and they then share the proceeds. The faster the players complete a room, the more money they earn. It is quite likely that the **players do not contribute equally** to the task. Some players probably have more experience than others, they know the subject matter matter for the puzzles in any given escape room, or they may just be more (less) interested in the game.



How should we split the reward between the players?

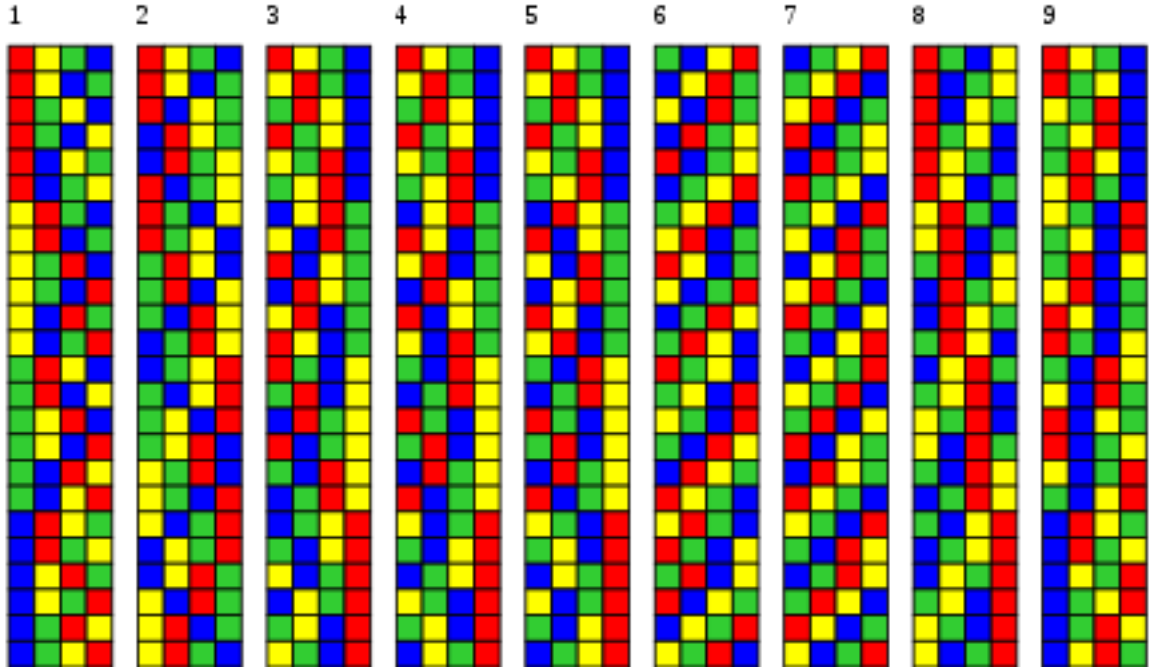
Shapley Value Example

Coalitions and Payoffs:

1. When playing alone:
   - A: $80
   - B: $56
   - C: $70

2. When playing in pairs:
   - A + B: $80
   - A + C: $85
   - B + C: $72

3. When all together:
   - A + B + C: $90



Shapley Value Example

Examples of Average Marginal Contribution:

- A plays alone = $80 (marginal contribution of A = $80)
- B joins A = $80 (marginal contribution of B = $0)
- C joins A and B = $90 (marginal contribution of C = $10)

- A plays alone = $80 (marginal contribution of A = $80)
- C joins A = $85 (marginal contribution of C = $5)
- B joins A and C = $90 (marginal contribution of B = $5)

Simply Put:

We look at all permuted sequences of players, and then average over the resulting marginal contributions.

Shapley Value Example

Average of Marginal Contributions:

- When we average over each player's marginal contributions (note, on the right, the marginal payouts are ordered by A, B, C)…

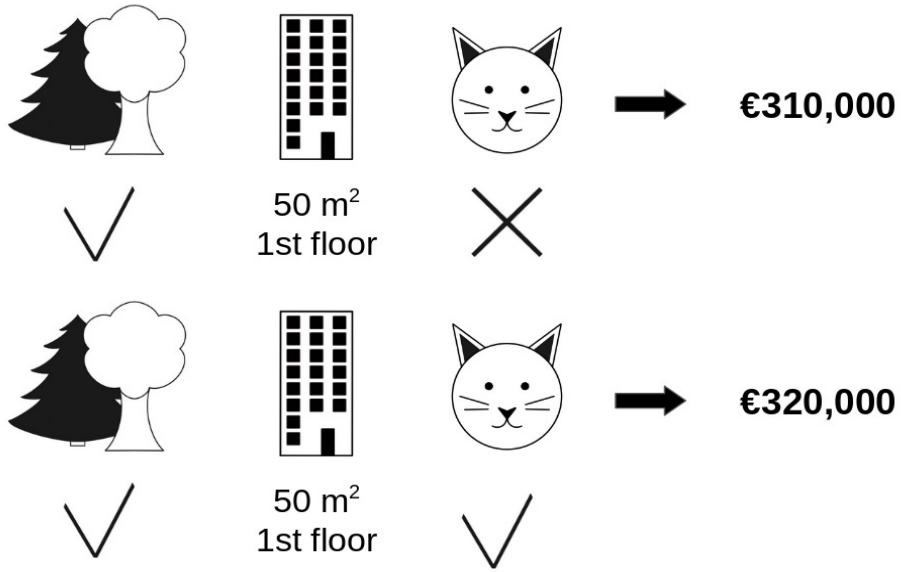
(A, B, C)	$(80, 0, 10)$
(A, C, B)	$(80, 5, 5)$
(B, A, C)	$(24, 56, 10)$
(B, C, A)	$(18, 56, 16)$
(C, A, B)	$(15, 5, 70)$
(C, B, A)	$(18, 2, 70)$

… we find that A contributes an average of (80 + 80 + 24 + 18 + 15 + 18) / 6 = 39.167, B contributes an average of 20.667, and C contributes an average of 30.167. These are our payoff weights, e.g., A contributes ~ twice as much as B.

SHAP: SHapley Additive exPlanations

Prediction Scenario:

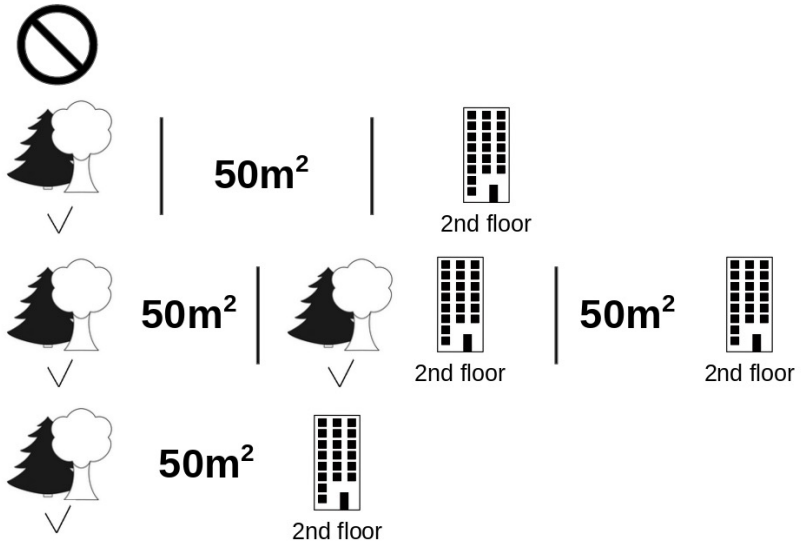
- We have a machine learning model that predicts some label.
- We have many games, i.e., the predictions the model produces for each observation.
- We thus have different coalitions of players, i.e., combinations of feature-*values*.
- Note that the average marginal contribution for a feature could be negative here.



SHAP: SHapley Additive exPlanations

Procedure – Marginal Contribution of 'Pets Allowed':

- We can cycle through all permutations of feature-values, just like in the escape room example.
- For a given feature value, we can randomly replace it with some other value from the data. Additionally, we can randomly replace other feature values with alternatives from the data.
- If we consider enough 'counterfactuals', we gain a sense of how 'moving across a feature's values' affects the marginal prediction, under many combinations of *other* feature values (partners).



SHAP: SHapley Additive exPlanations

Marginal Contribution of a Pixel:

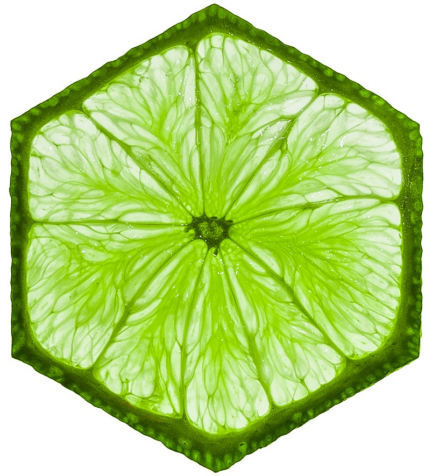
- In a CNN, our input predictors are pixels. We can do the same thing here using pixel values.
- We can figure out a Shapley value for a particular pixel position, in a particular color channel, for feature values that range from 0 to 255.
- To simplify the problem, we might also work with super-pixels (group pixels, averaging their feature-values).

Local Interpretable Model-agnostic Explanations

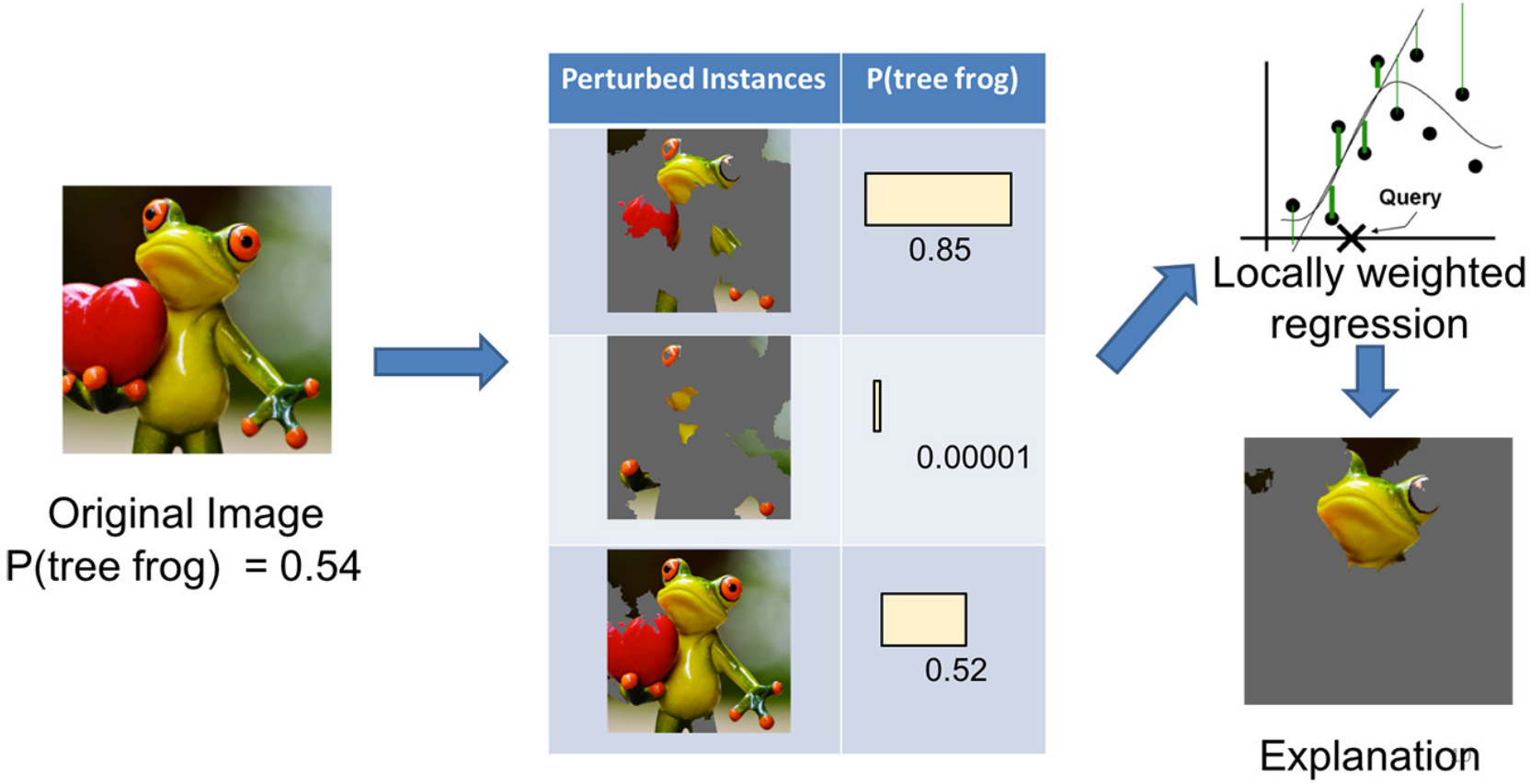
Procedure – Local Perturbations of the Data:

- We assume that a features' contributions to the prediction can be locally approximated by a linear regression. Then:

    1. For a given prediction, **randomly perturb the observation** (modify its feature values), repeatedly, and **recover the associated predictions** for each synthetic observation.
    2. This procedure **yields an observation-specific dataset**. Use the dataset to **estimate a weighted linear regression** on the dataset, weighting observations inversely by their distance / dissimilarity from the original observation.
    3. **Beta coefficients** reflect features' localized marginal contributions to the prediction.



Local Interpretable Model-agnostic Explanations



Questions?