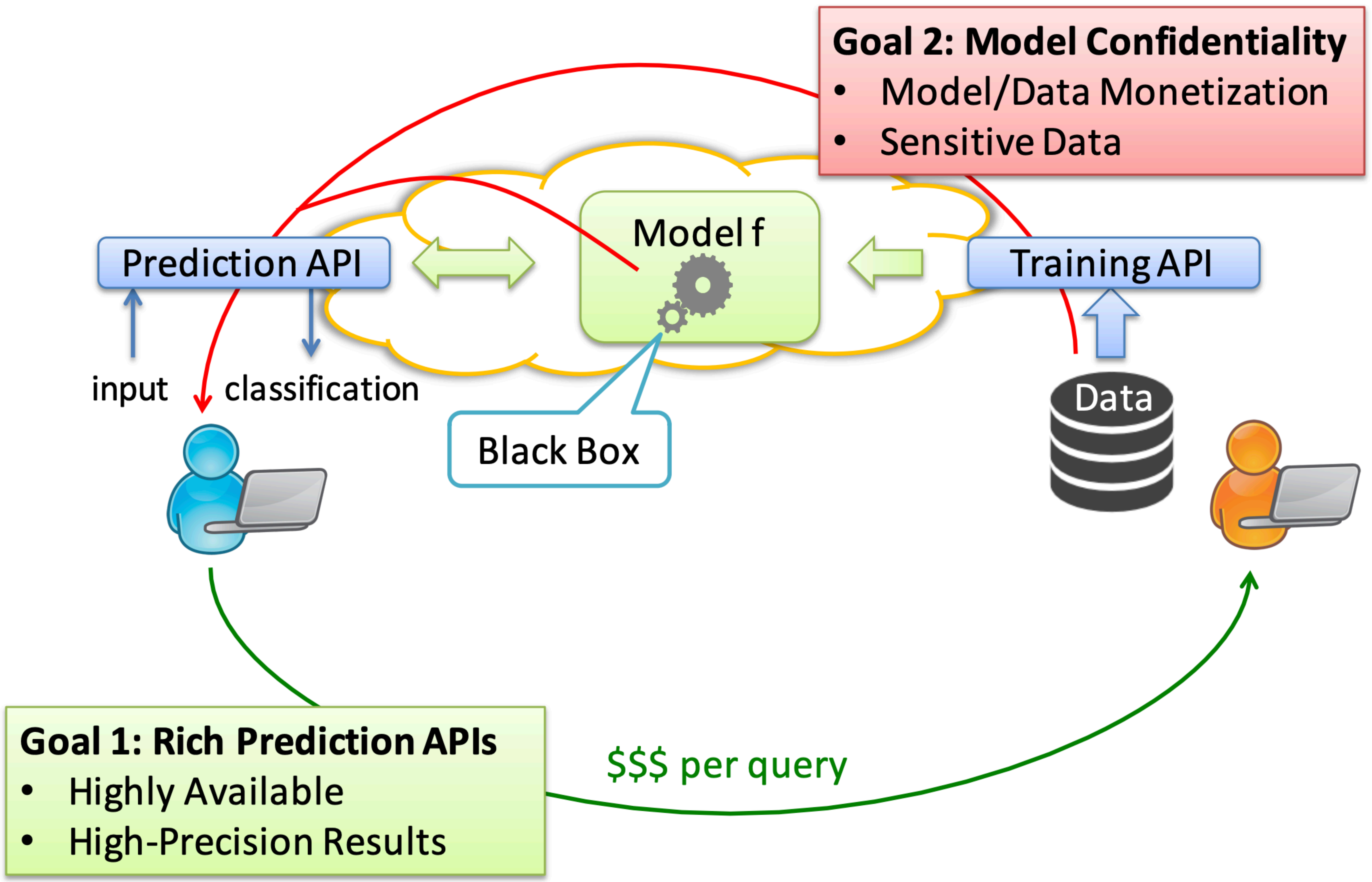
COMP6211I: Trustworthy Machine Learning

Model Confidentiality (attack)

Minhao CHENG

Machine learning as a service (MaaS)



Attack Taxonomy

- Theft
  - Accuracy
- Reconnaissance
  - Fidelity
  - Function Equivalence

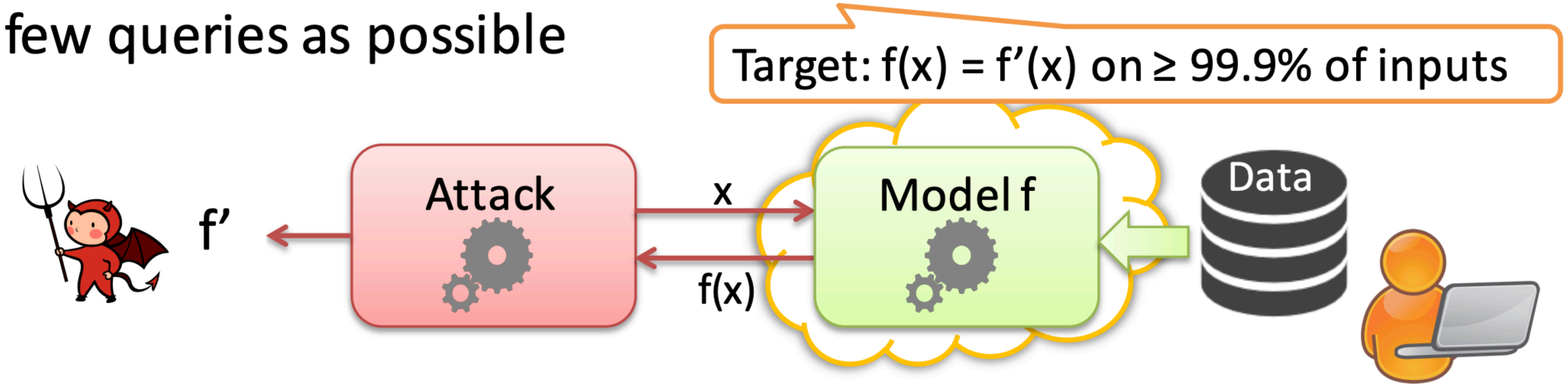
Threat model

- Could only query the model with confidence output
- No idea about the training procedure
- Model architecture

Learning based extraction

Model extraction attack

Goal: Adversarial client learns close approximation of f using as few queries as possible



Applications:

1) Undermine pay-for-prediction pricing model
2) Facilitate privacy attacks (
3) Stepping stone to model-evasion
   [Lowd, Meek – 2005] [Srndic, Laskov – 2014]

Learning based extraction

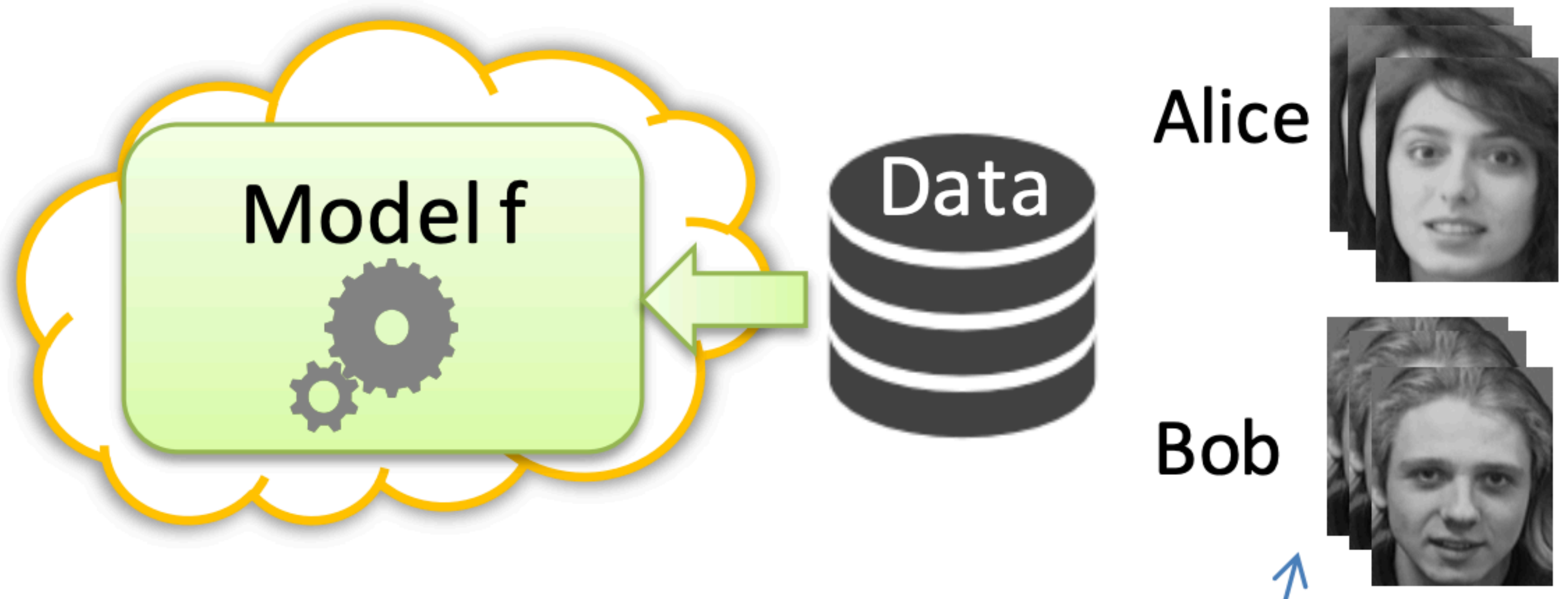
Model extraction example: Logistic regression

Task: Facial Recognition of two people (binary classification)

n+1 parameters w,b chosen using training set to minimize expected error

$$f(x) = 1 / (1+e^{-(w*x + b)})$$

f maps features to predicted probability of being "Alice"
≤ 0.5 classify as "Bob"
> 0.5 classify as "Alice"



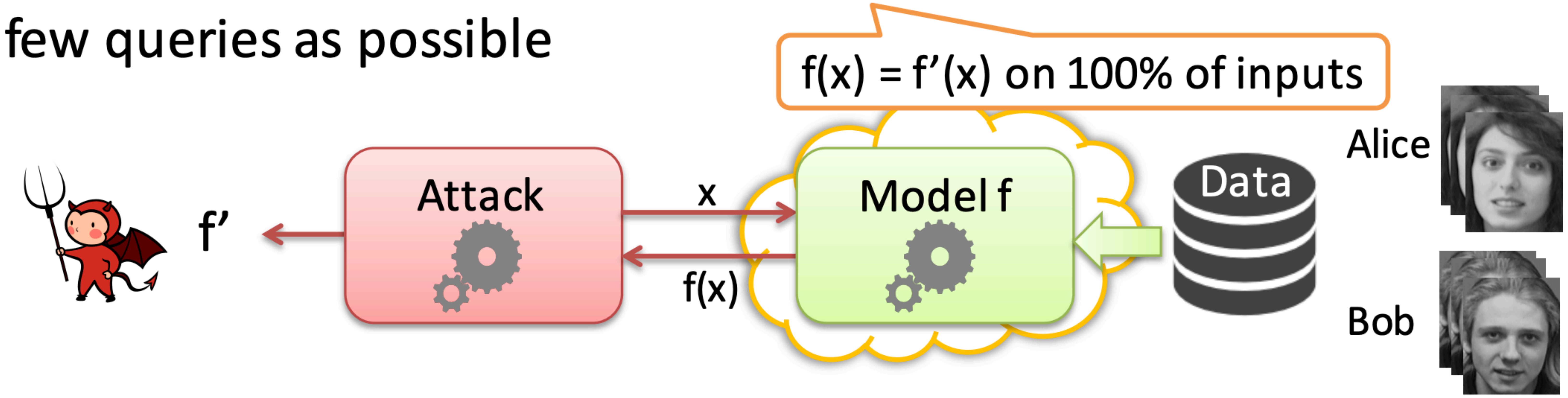
Feature vectors are pixel data
e.g., $n = 92 * 112 = 10{,}304$

Generalize to c > 2 classes with *multinomial logistic regression*
$f(x) = [p_1, p_2, \ldots, p_c]$ predict label as $\text{argmax}_i\, p_i$

Learning based extraction

Model extraction example: Logistic regression

Goal: Adversarial client learns close approximation of f using as few queries as possible



$$f(x) = 1 / (1+e^{-(w*x + b)})$$

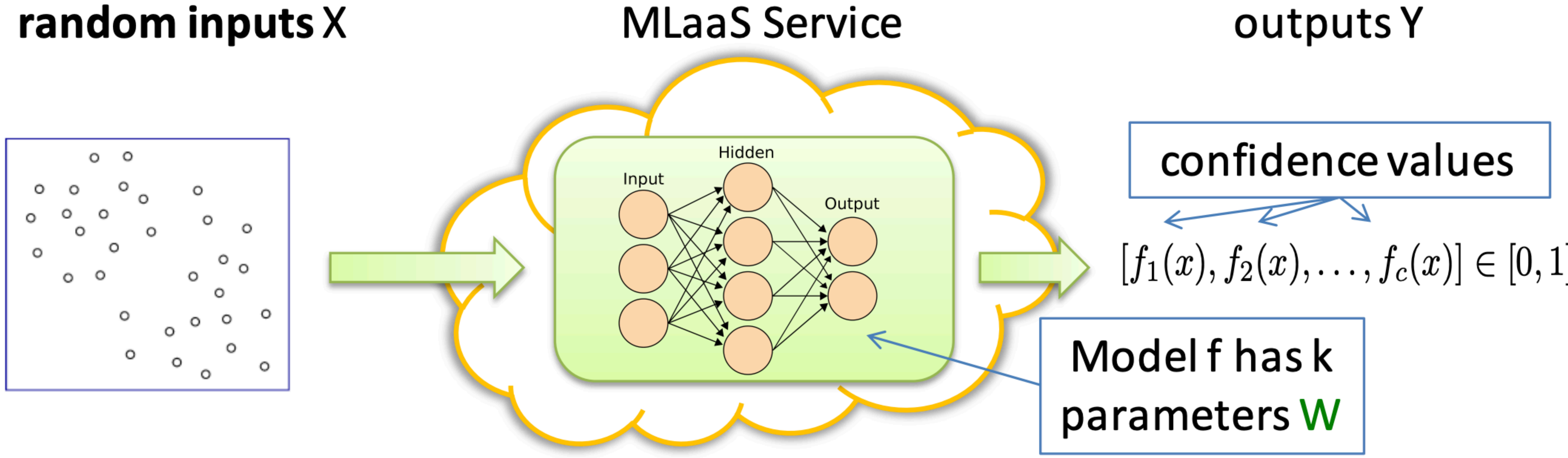
$$\ln\left(\frac{f(x)}{1 - f(x)}\right) = w*x + b$$

Linear equation in n+1 unknowns w,b

Query n+1 random points ⇒ solve a linear system of n+1 equations

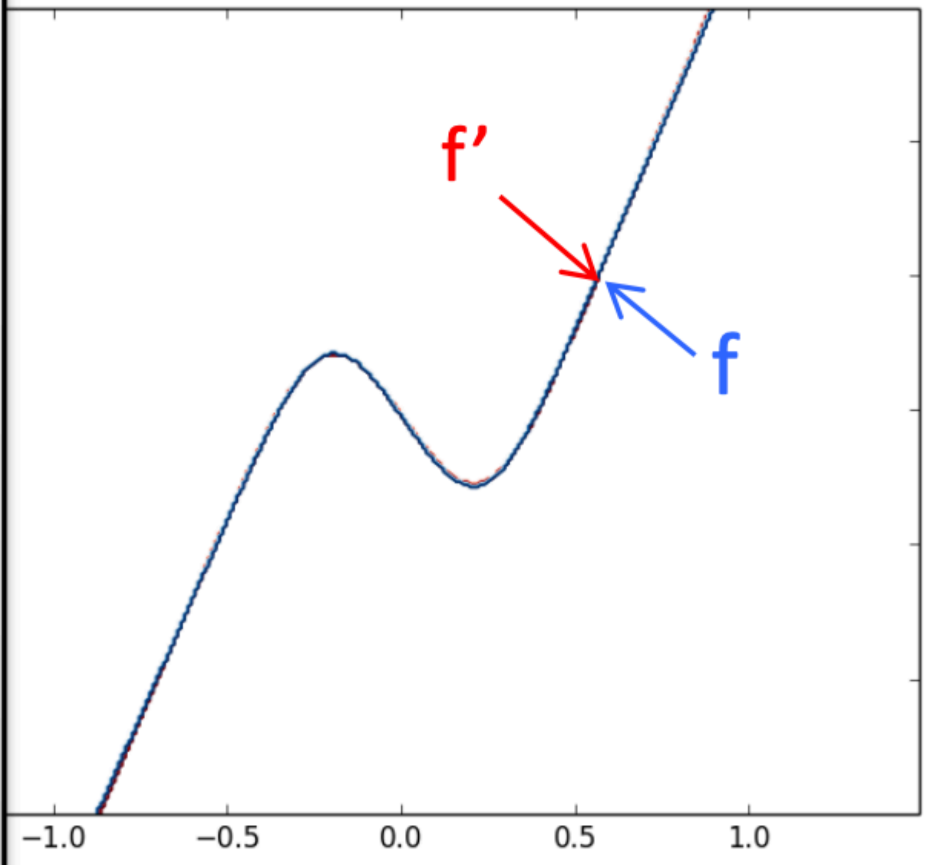
Learning based extraction

Generic equation-solving attack



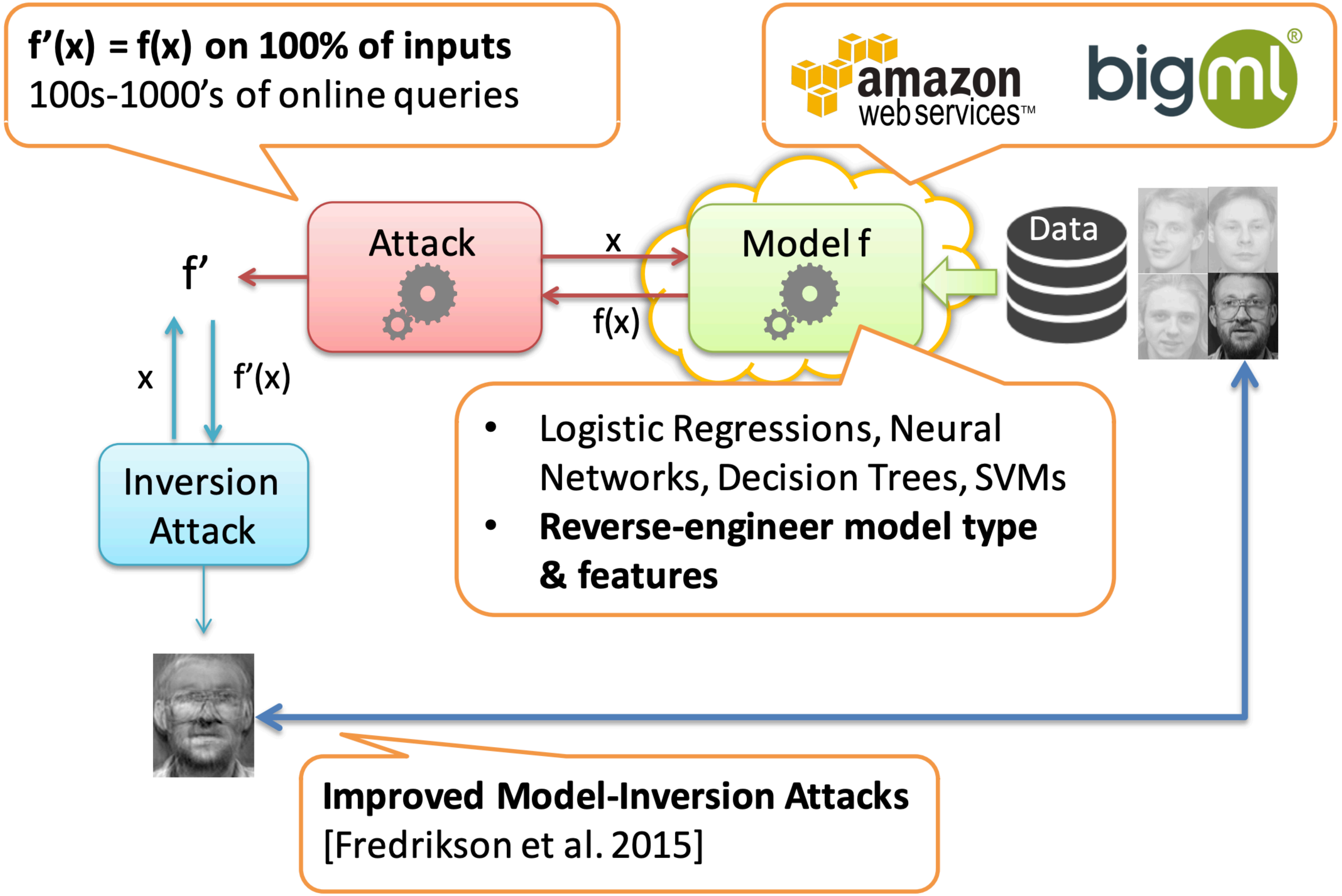
$$[f_1(x), f_2(x), \ldots, f_c(x)] \in [0,1]^c$$

- Solve non-linear equation system in the weights W
  - Optimization problem + gradient descent
  - *"Noiseless Machine Learning"*
- Multinomial Regressions & Deep Neural Networks:
  - >99.9% agreement between f and f'
  - ≈ 1 query per model parameter of f
  - 100s - 1,000s of queries / seconds to minutes



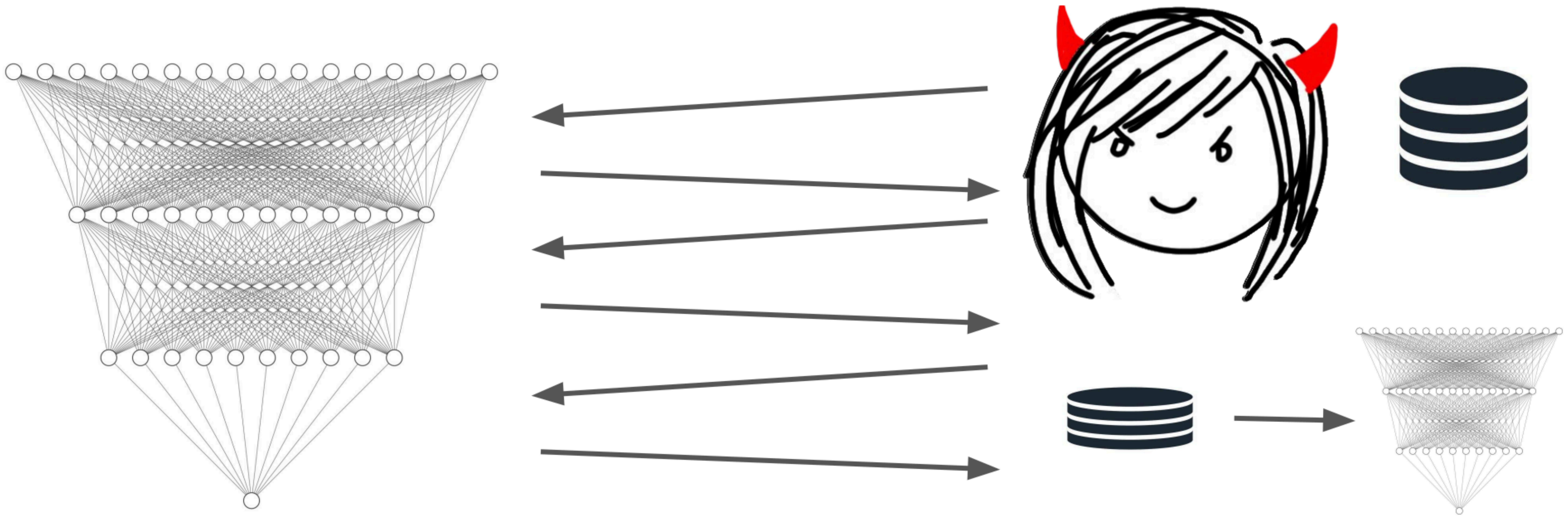
Learning based extraction

Combination of model inversion



Learning based extraction

Improvements: active learning



Active Learning: progressively growing a labeled dataset

Chandrasekharan et al: https://arxiv.org/abs/1811.02054

Learning based extraction

Improvements: semi-supervised learning

- Augments the model with rotation loss
  - Labeled data: The classifier
  - Unlabeled data: The rotation loss

$$L_R(X; f_\theta) = \frac{1}{4N} \sum_{i=0}^{N} \sum_{j=1}^{r} H(f_\theta(R_j(x_i)), j)$$

Learning based extraction

Results

- Semi-supervised learning
  - Scales to deep learning + complex datasets
  - Requires large unlabeled dataset
- Label efficient!

Dataset	Queries	Baseline Accuracy	SemiSup Accuracy
SVHN	250	79.25%	95.82%
CIFAR-10	250	53.35%	87.98%
ImageNet (top 5)	~140000	83.5%	86.17%

Learning based extraction

Limitations

- Yields high accuracy model but …
- Not high fidelity
- High fidelity:
  - Both correct and wrong
  - Better to be used in substitute model
    - Adversarial attack
    - Model inversion attack
    - …

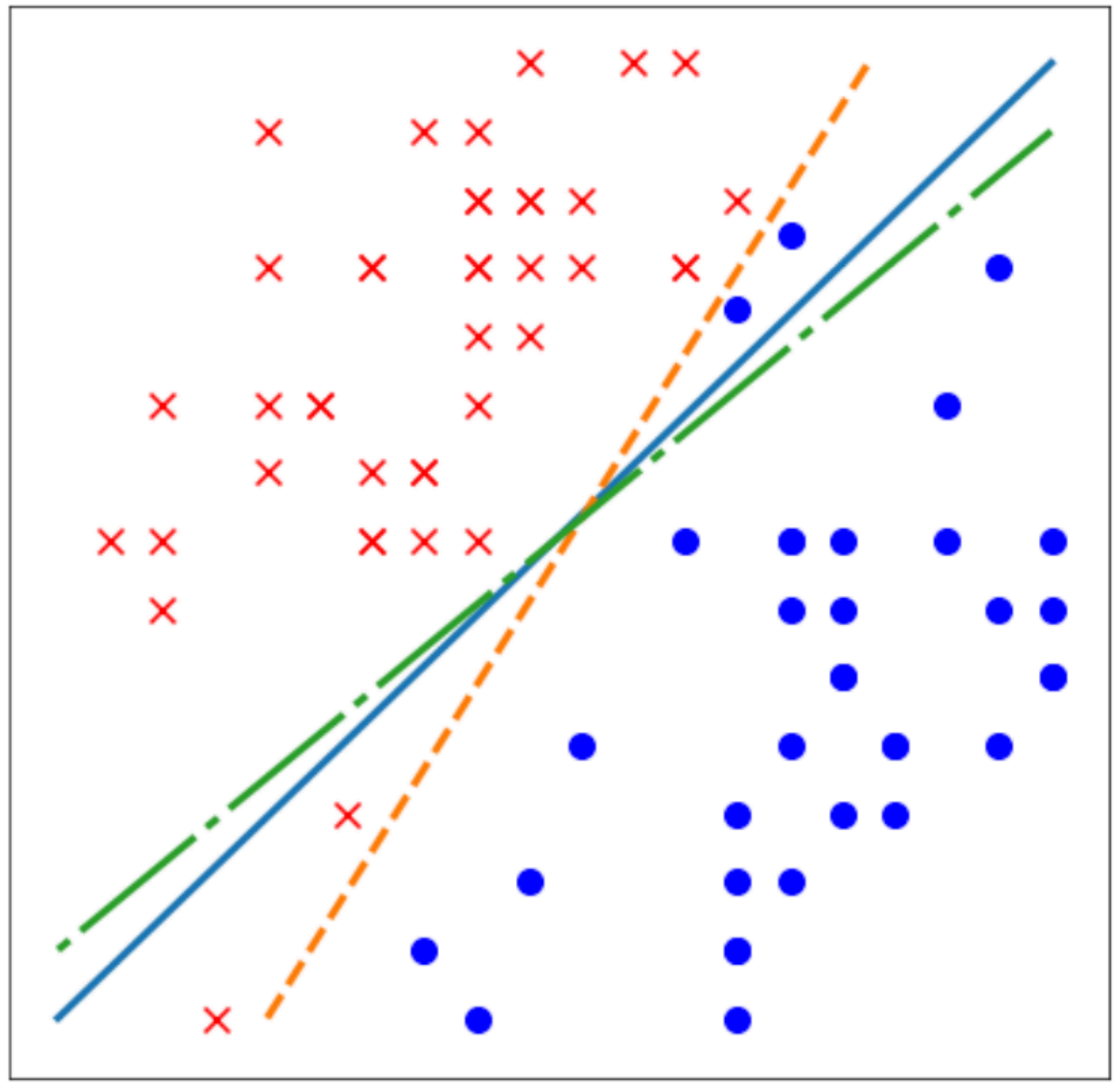
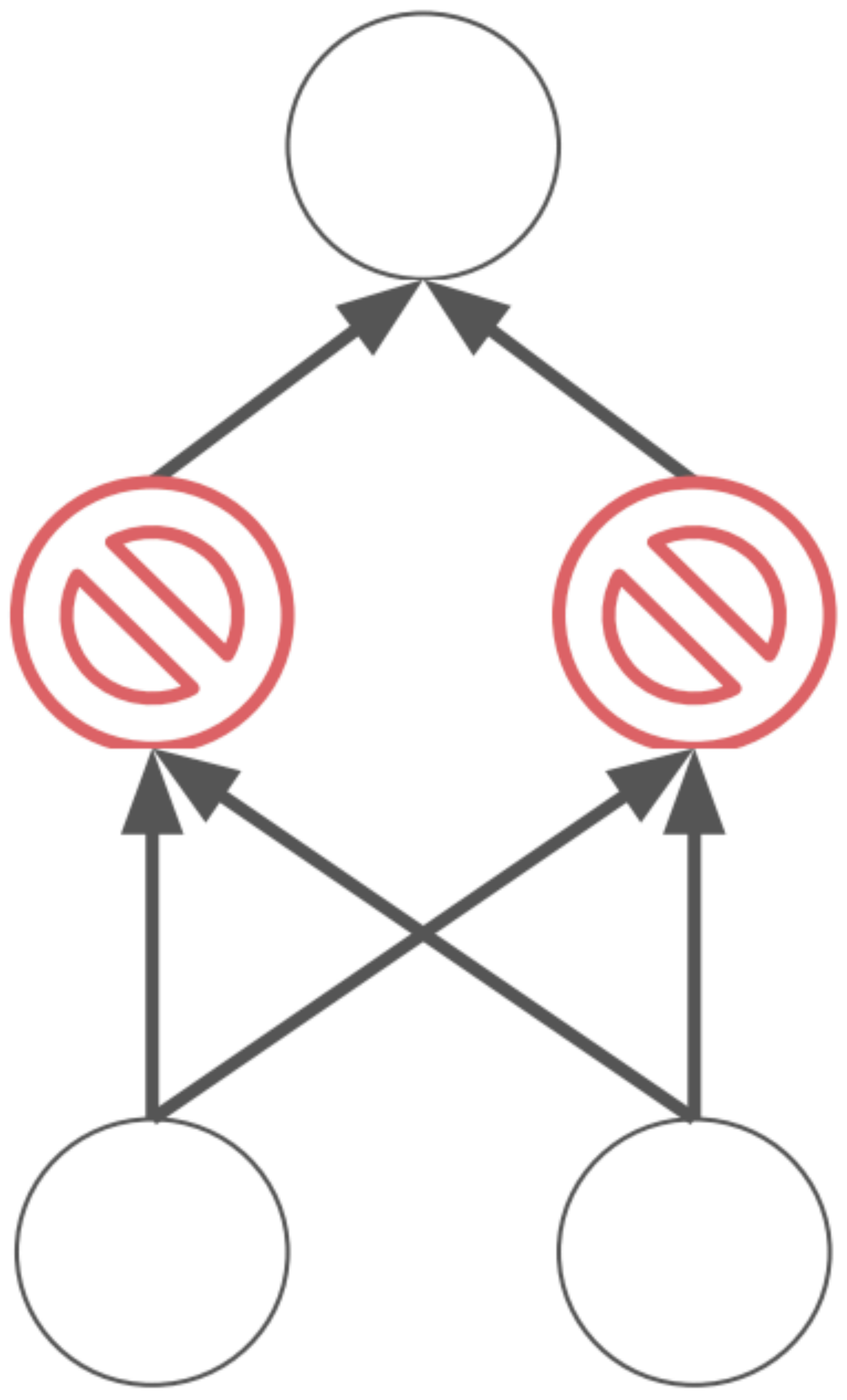
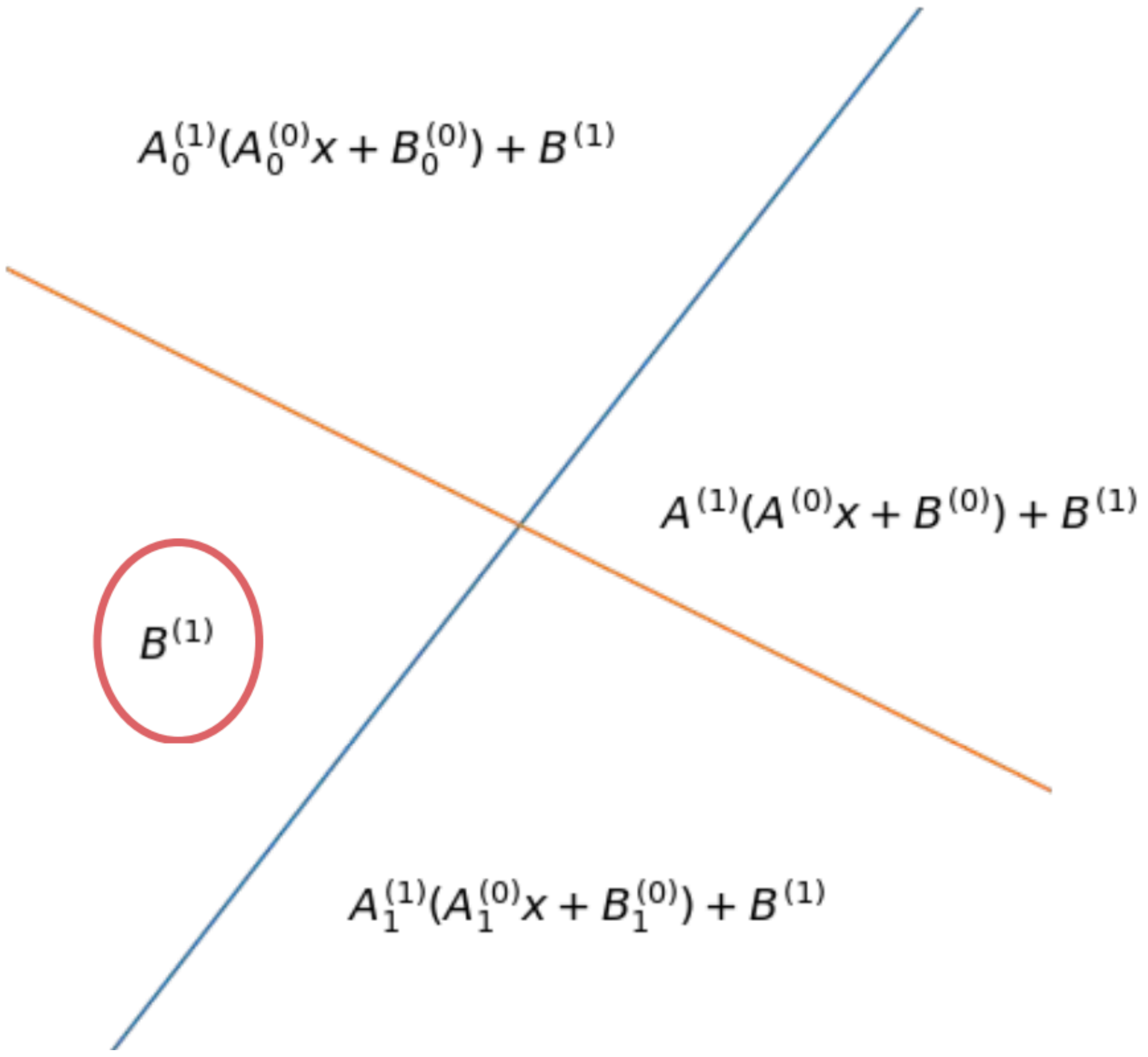


Figure 1: Illustrating fidelity vs. accuracy. The solid blue line is the oracle; functionally equivalent extraction recovers this exactly. The green dash-dot line achieves high fidelity: it matches the oracle on all data points. The orange dashed line achieves perfect accuracy: it classifies all points correctly.

Function equivalent extraction

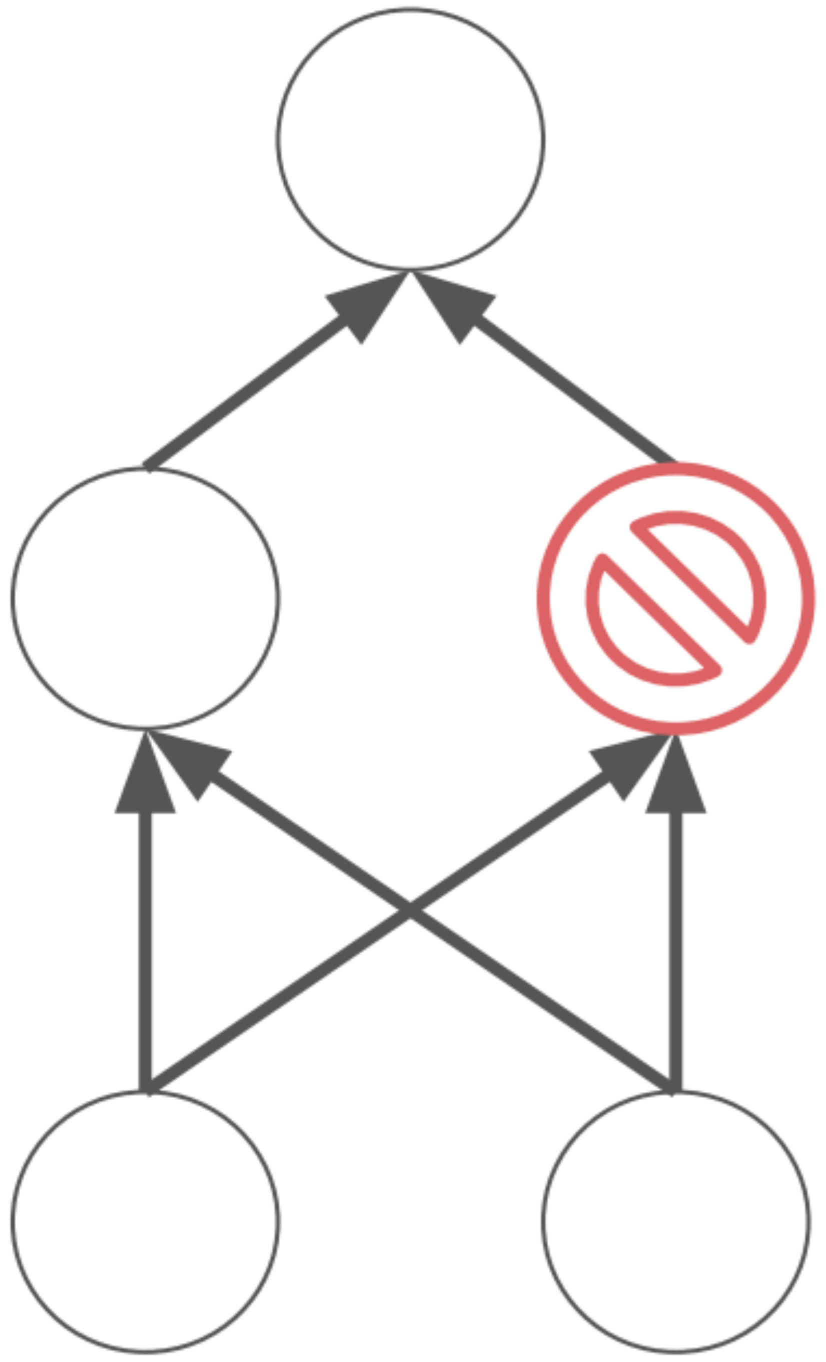
Intuition

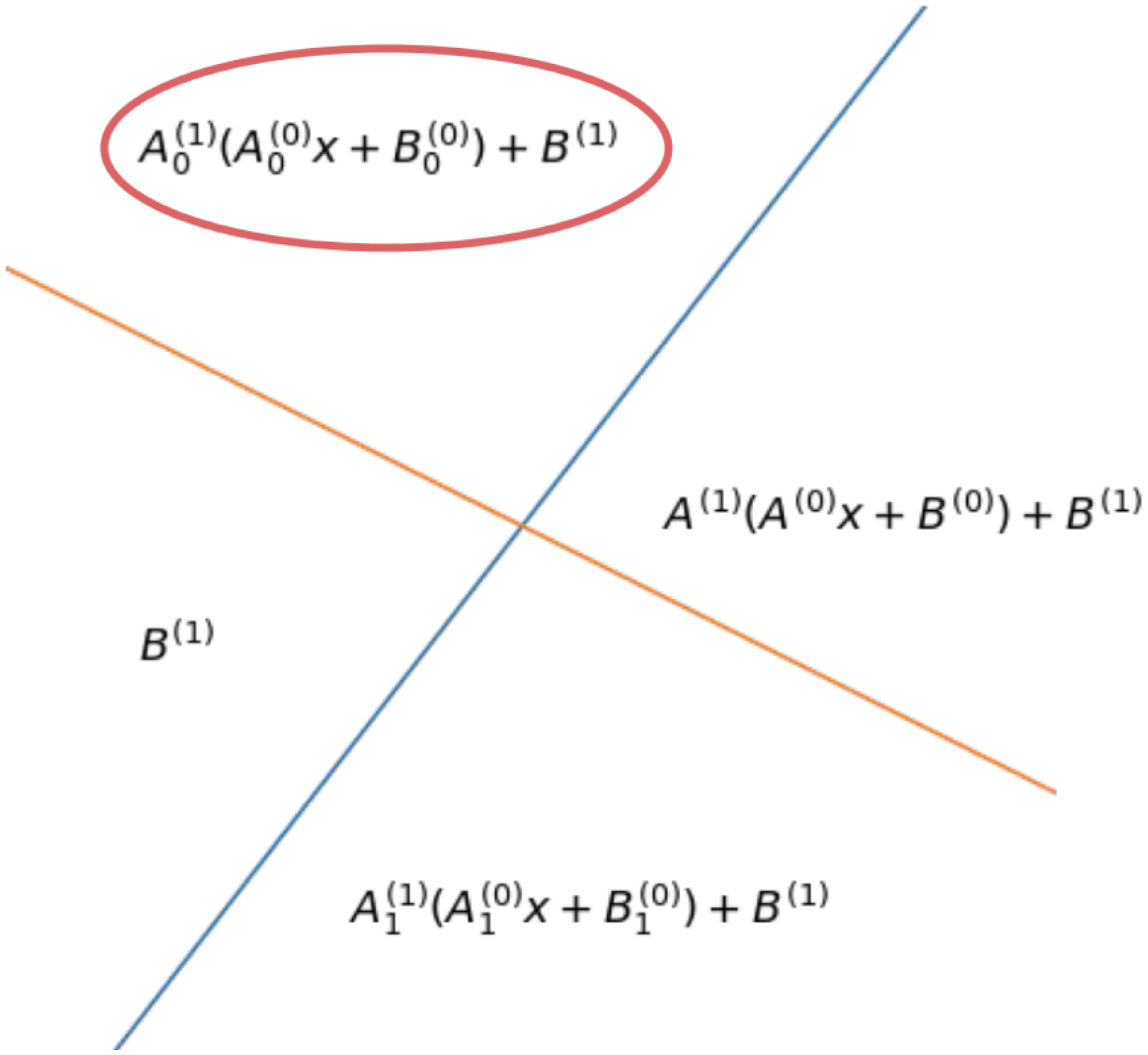




Function equivalent extraction

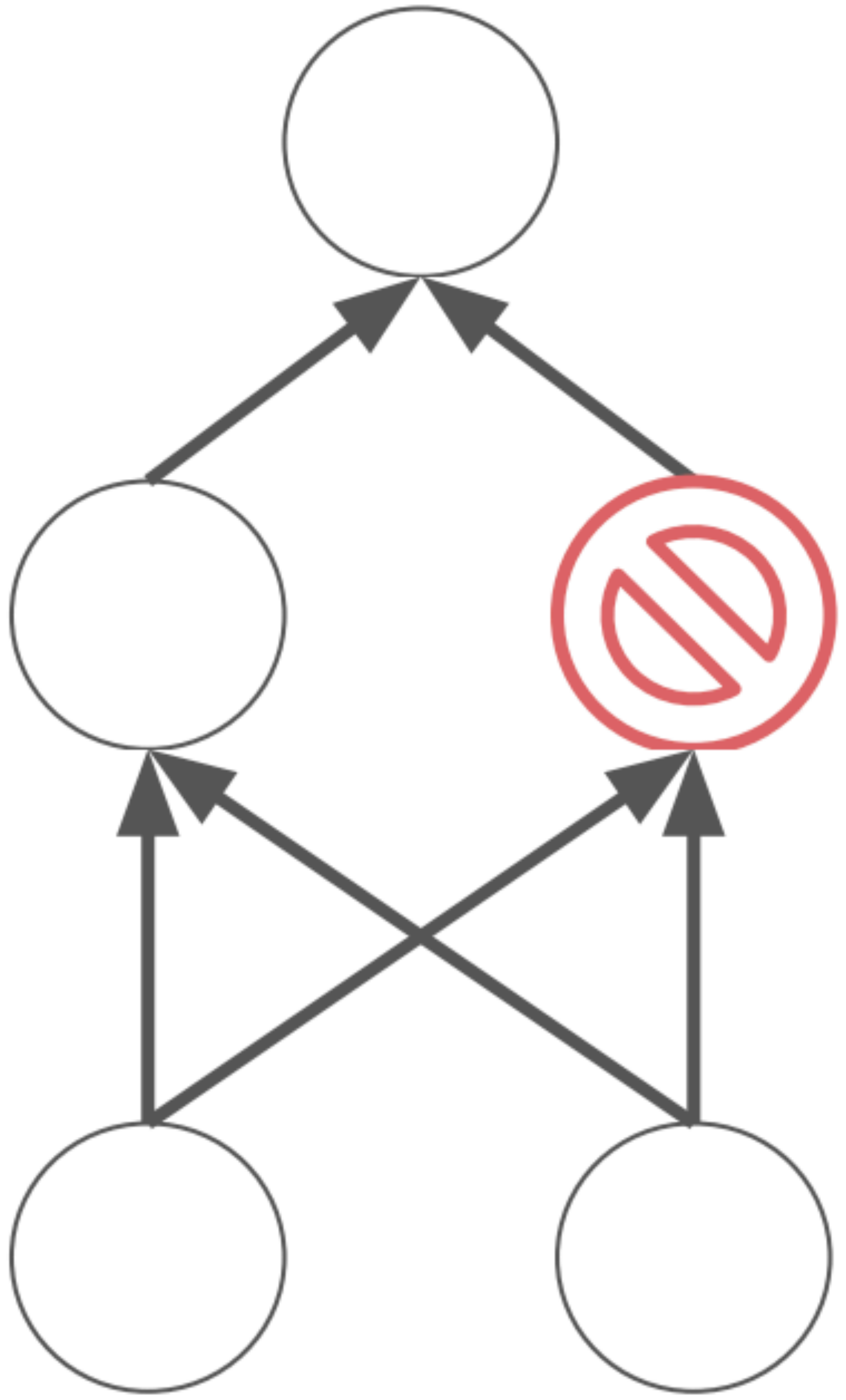
Intuition

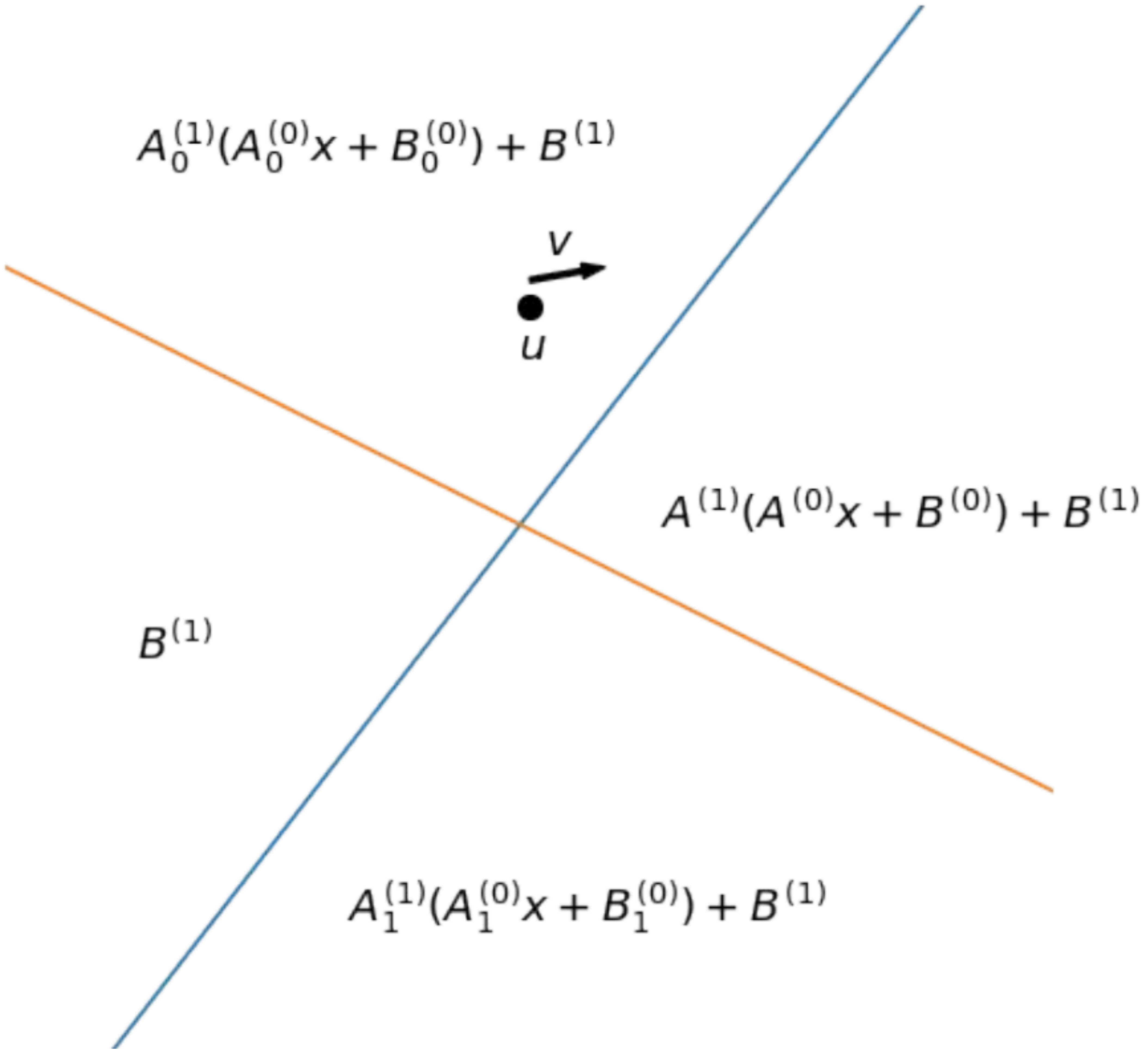




Function equivalent extraction

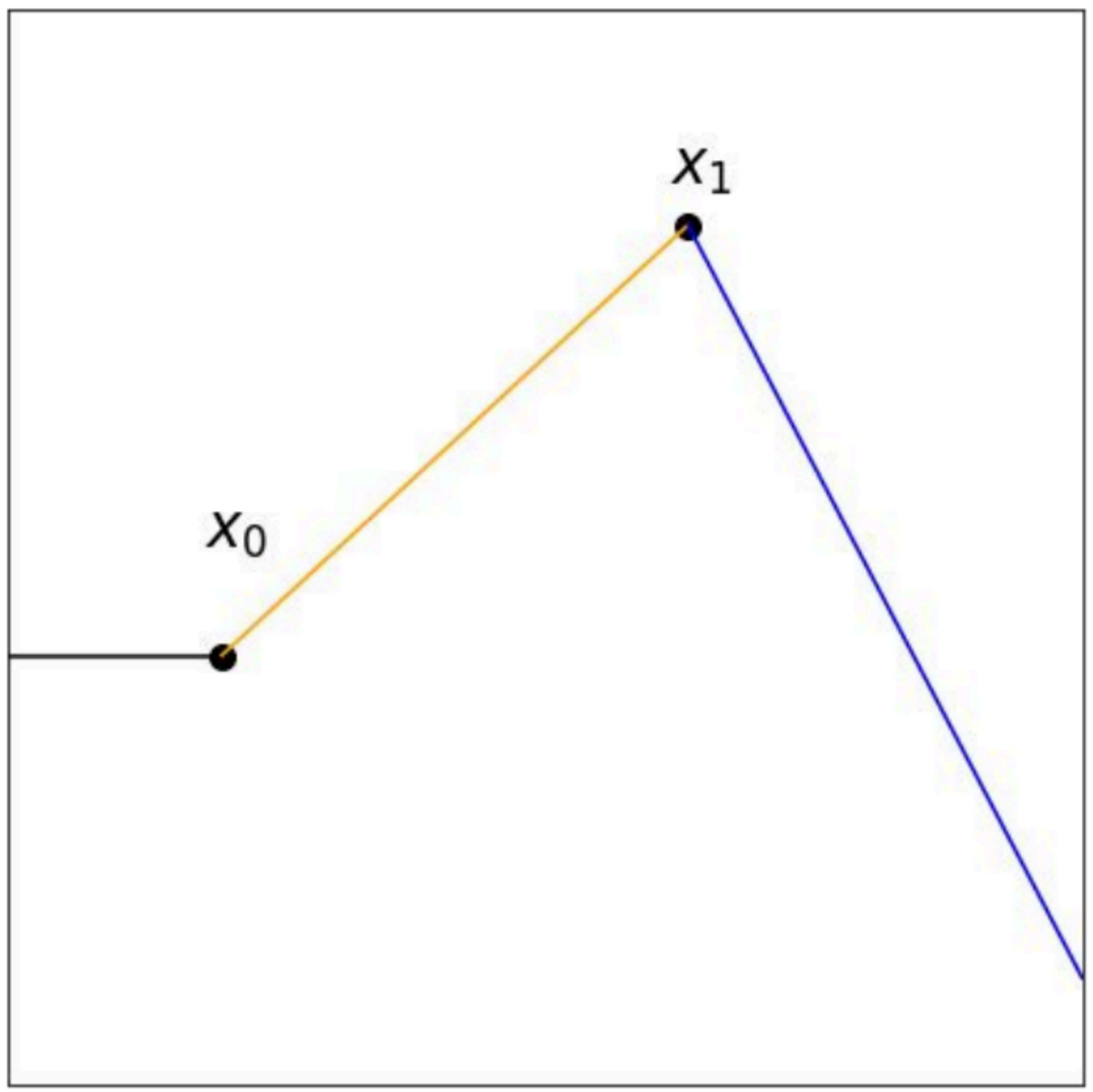
Intuition

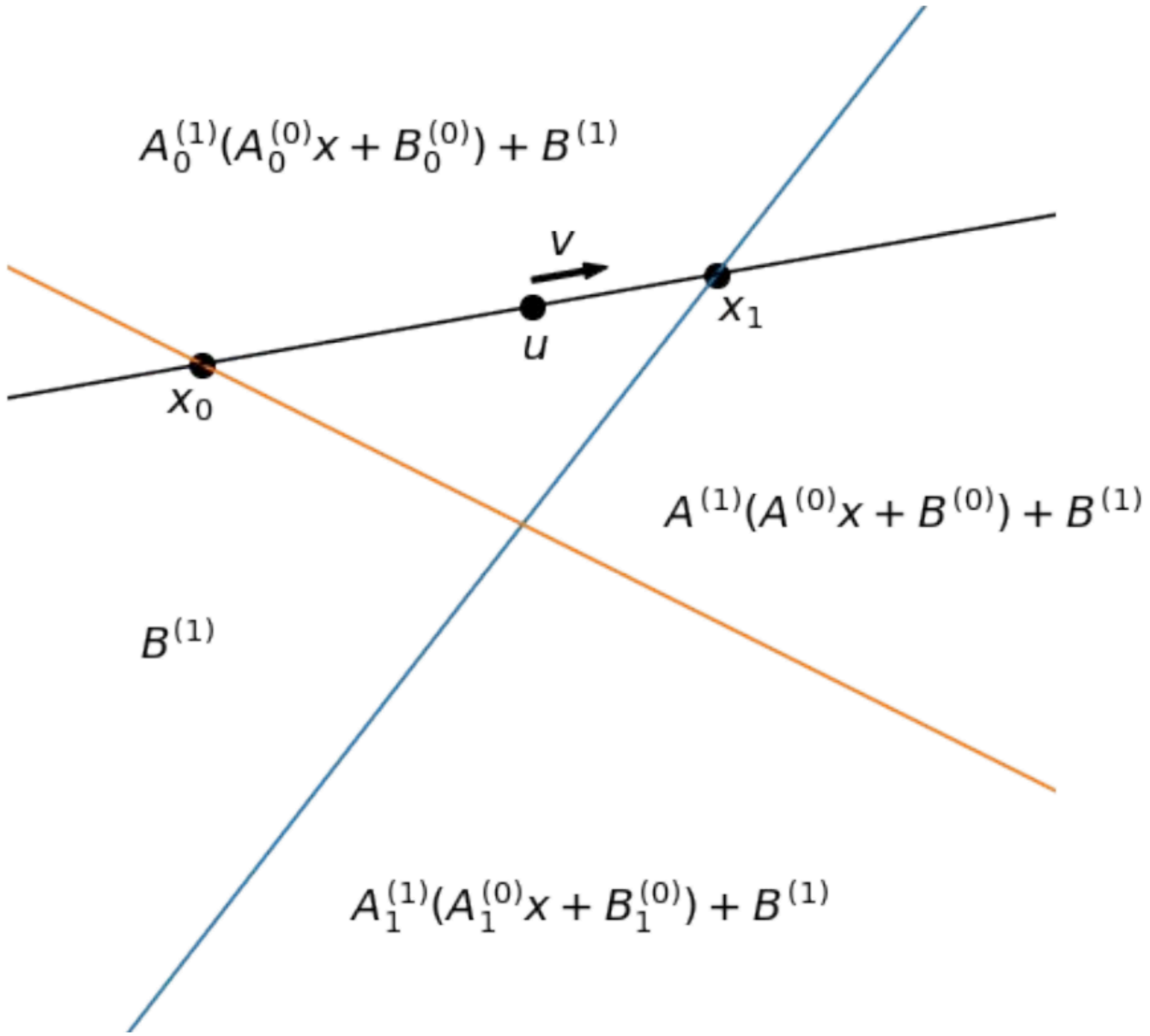




Function equivalent extraction

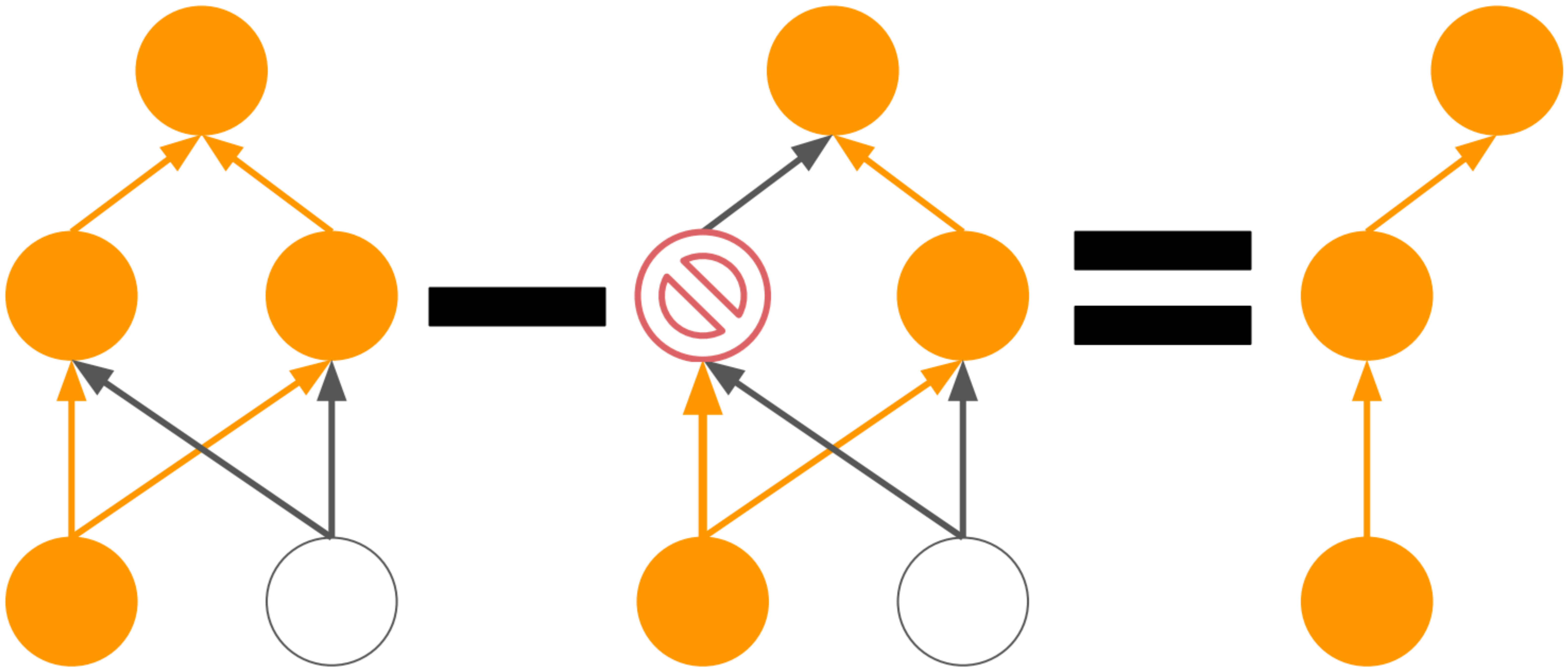
Intuition





Function equivalent extraction

Intuition



Function equivalent extraction

- Critical point search
  - Identify $\{x_i\}_{i=1}^{n}$ exactly one of the ReLU units is at a critical point
- Weight recovery
- Sign recovery
- Final layer extraction

Function equivalent extraction

Critical point search

- For two layer neural networks:
    - $O_L(x) = A^{(1)}\text{ReLU}(A^{(0)}x + B^{(0)}) + B^{(1)}.$
- To find a critical point
    - $L(t; u, v, O_L) = O_L(u + tv).$
    - Not differential -> some ReLU change signs
    - Problem: not efficient

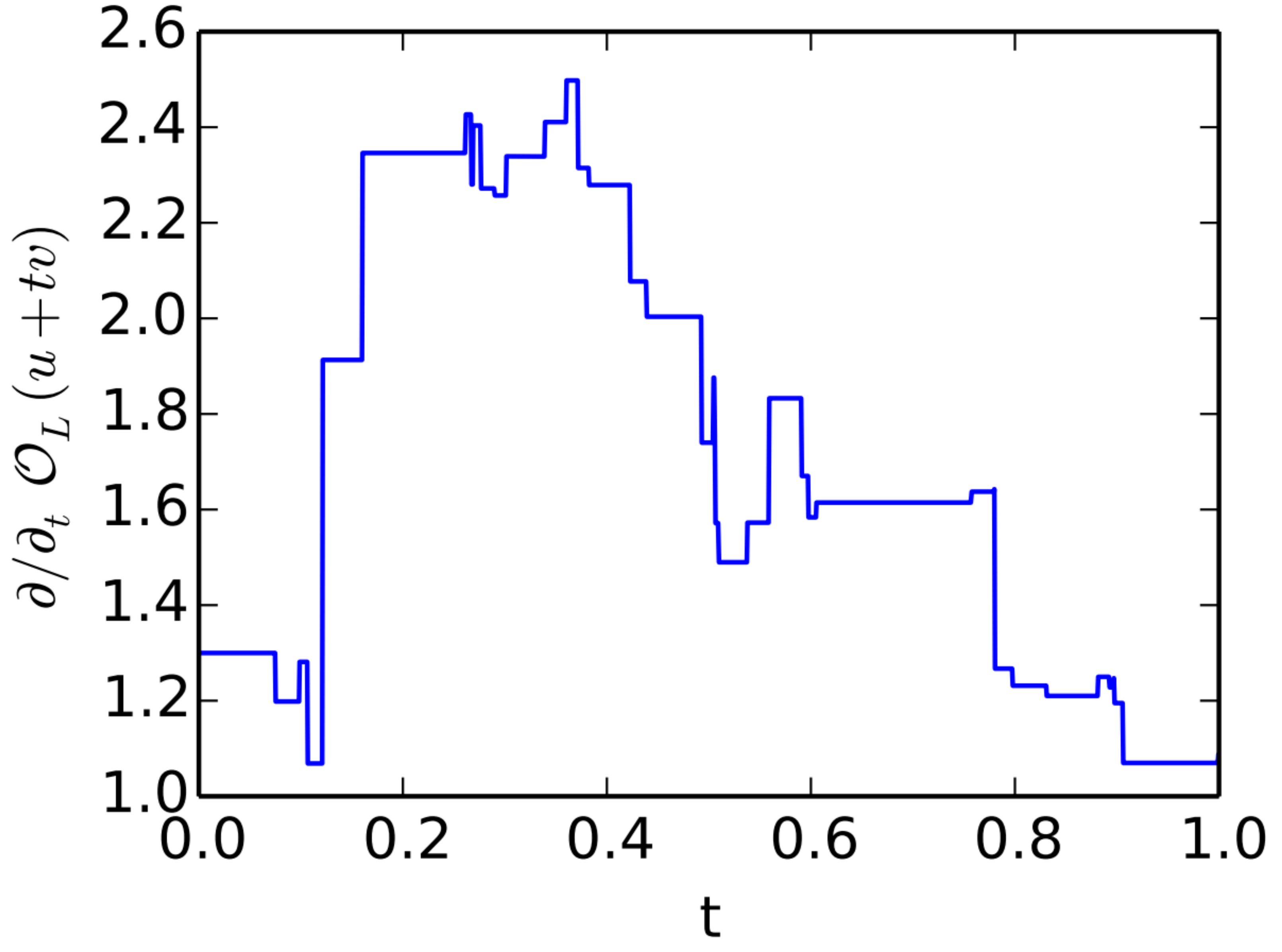


Figure 3: An example sweep for critical point search. Here we plot the partial derivative across t and see that $O_L(u+tv)$ is piecewise linear, enabling a binary search.

Function equivalent extraction

2-linear testing subroutine

- If the range is composed by two line segments
  - Identify the linear segment
  - Compute the intersection

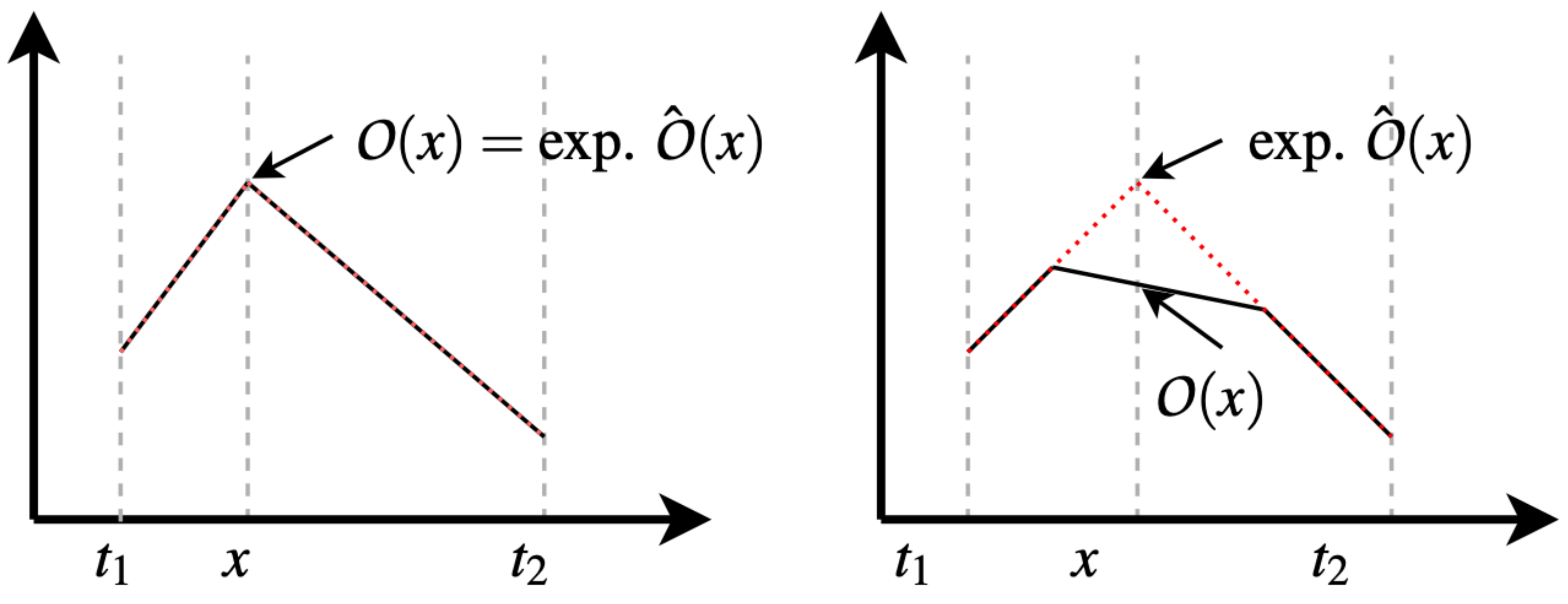


Figure 4: Efficient and accurate 2-linear testing subroutine in Algorithm 1. Left shows a successful case where the algorithm succeeds; right shows a potential failure case, where there are multiple nonlinearities. We detect this by observing the expected value of $O(x)$ is not the observed (queried) value.

Function equivalent extraction

Weight recovery

- For a critical point x_i, and a random input-space direction e_j

$$
\begin{aligned}
\left.\frac{\partial^2 O_L}{\partial e_j^2}\right|_{x_i} &= \left.\frac{\partial O_L}{\partial e_j}\right|_{x_i+c\cdot e_j} - \left.\frac{\partial O_L}{\partial e_j}\right|_{x_i-c\cdot e_j} \\
&= \sum_k A_k^{(1)} \mathbb{1}(A_k^{(0)}(x_i + c\cdot e_j) + B_k^{(0)} > 0) A_{kj}^{(0)} \\
&\quad - \sum_k A_k^{(1)} \mathbb{1}(A_k^{(0)}(x_i - c\cdot e_j) + B_k^{(0)} > 0) A_{kj}^{(0)} \\
&= A_i^{(1)} \Big(\mathbb{1}(A_i^{(0)} \cdot e_j > 0) - \mathbb{1}(-A_i^{(0)} \cdot e_j > 0) \Big) A_{ji}^{(0)} \\
&= \pm (A_{ji}^{(0)} A_i^{(1)})
\end{aligned}
$$

Function equivalent extraction

Weight recovery

- With e_1 and e_2,
  - We could compute $|A_{1i}^{(0)}A^{(1)}|$ and $|A_{2i}^{(0)}A^{(1)}|$
  - Then we could get $|A_{1i}^{(0)}/A_{2i}^{(0)}|$
- We can get $|A_{1i}^{(0)}/A_{ki}^{(0)}|$ for all k
- Just assign $A_{1i}^{(0)} = 1$

$$
\begin{aligned}
\left.\frac{\partial^2 O_L}{\partial e_j^2}\right|_{x_i} &= \left.\frac{\partial O_L}{\partial e_j}\right|_{x_i+c\cdot e_j} - \left.\frac{\partial O_L}{\partial e_j}\right|_{x_i-c\cdot e_j} \\
&= \sum_k A_k^{(1)} \mathbb{1}(A_k^{(0)}(x_i + c\cdot e_j) + B_k^{(0)} > 0) A_{kj}^{(0)} \\
&\quad - \sum_k A_k^{(1)} \mathbb{1}(A_k^{(0)}(x_i - c\cdot e_j) + B_k^{(0)} > 0) A_{kj}^{(0)} \\
&= A_i^{(1)} \left(\mathbb{1}(A_i^{(0)} \cdot e_j > 0) - \mathbb{1}(-A_i^{(0)} \cdot e_j > 0) \right) A_{ji}^{(0)} \\
&= \pm (A_{ji}^{(0)} A_i^{(1)})
\end{aligned}
$$

Function equivalent extraction

Weight sign recovery

- For a critical point x_i in the direction $e_j + e_k$

$$\left.\frac{\partial^2 O_L}{\partial(e_j + e_k)^2}\right|_{x_i} = \pm(A_{ji}^{(0)} A_i^{(1)} \pm A_{ki}^{(0)} A_i^{(1)}).$$

- As we know the scale,
  - Just to check the gradient is cancelled or compounded

Function equivalent extraction

Last layer recover

- After got the first layer, the logit function is a linear transformation
- Recover by least square
  - With the critical point to save # of queries

Function equivalent extraction

Results

Parameters	25,000	50,000	100,000
Fidelity	100%	100%	99.98%
Queries	~150,000	~300,000	~600,000

Effectiveness of our Direct Recovery Attack

Counter measurements

Hard label output

How to prevent extraction?

API Minimization

$f(x) = y$

Prediction

~~Confidence~~

- Prediction = class label only
- *Learning with Membership Queries*

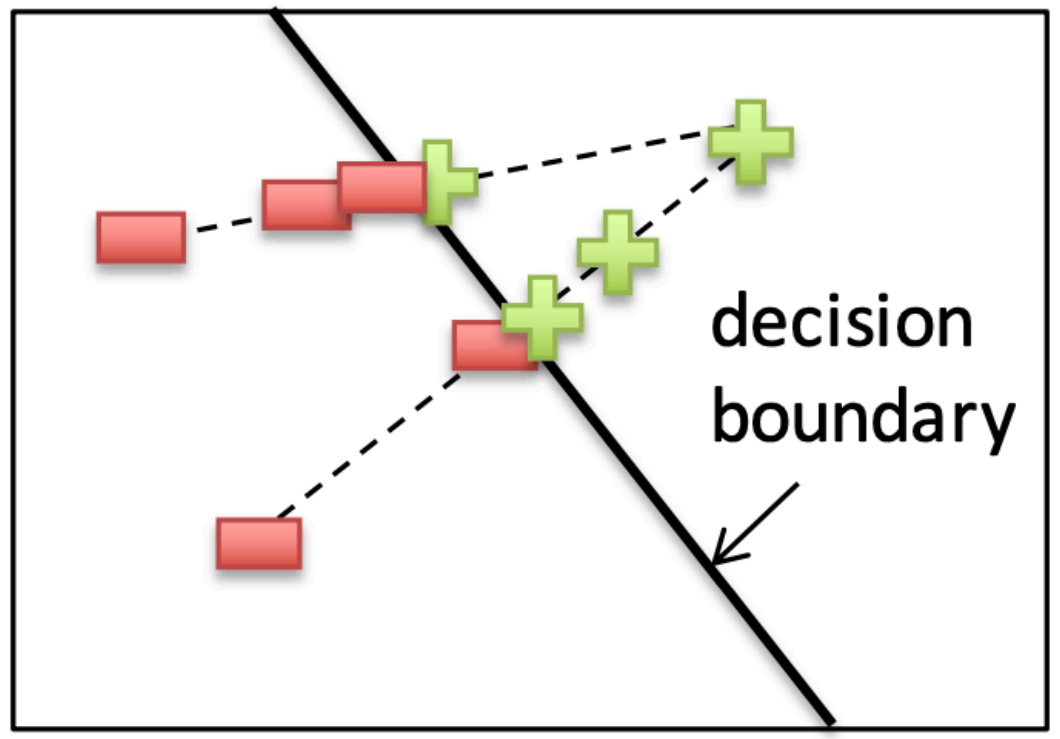
Attack on Linear Classifiers [Lowd,Meek – 2005]

classify as "+" if $w*x + b > 0$ and "-" otherwise

n+1 parameters w,b

$f(x) = \text{sign}(w*x + b)$

1. Find points on decision boundary ($w*x + b = 0$)
   - Find a "+" and a "-"
   - Line search between the two points
2. Reconstruct w and b (up to scaling factor)



Counter measurements

- In the next class
  - Make the feature unlearnable
- DP will cover later in the course