Mean Teacher are better role models

Weight-averaged consistency targets improve semi-supervised deep learning results

Antti Tarvainen & Harri Valpola (Curious AI)

NIPS 2017

Talk Highlights

- Semi-supervised learning (Why Deep ?)
- Taxonomy
- Key contributions and takeways
- Methodology
- Results
- Conclusion

Semi-supervised learning(Why deep?)

- Use both labeled and unlabeled data
- Not especially constructed for neural networks
  - Treats them as yet another supervised model, whose intermediate representations can be updated
  - Ensemble of models provides more robust predictions
    - Using different samples of same dataset via Bootstrap Sampling
    - Same model but different feature sets
    - Different classifiers with different inductive bias
    - Disagreement based models
- When applied to neural networks, the implementation only varies
  - Adding noise or pertubing the input data
  - Perturbing the feature representation
  - Using llabel predictions or activations or model weights

Taxonomy : semi-supervised

Self training

Multi-view training
- Co-training
- Democratic Co-learning
- Tri-training
- Tri-training with disagreement
- Asymmetric tri-training
- Multi-task tri-training

Self ensembling
- Ladder networks
- Virtual Adversarial Training
- Π model
- Temporal Ensembling
- Mean Teacher

Other :
- Distillation
- Learning from weak supervision
- Learning with noisy labels
- Data augmentation
- Ensembling a single model

Self-ensembling : Makes use of a single model under different configurations in order to create a stronger ensemble prediction.

1. *Invariant to noise* : Uses model prediction on clean example as a proxy label for prediction on a perturbed version(Ladder networks) or perturb in the feature space (Virtual Adversarial Training)
2. Ensemble the predictions of the model under two different perturbations(noise) of input data or different dropout conditions
3. Ensemble the predictions of the model at different timesteps (after each epoch or iteration)

S. Ruder, An overview of proxy-label approaches for semi-supervised learning, *http://ruder.io/semi-supervised/index.html#selfensembling*

Key takeaways and contributions

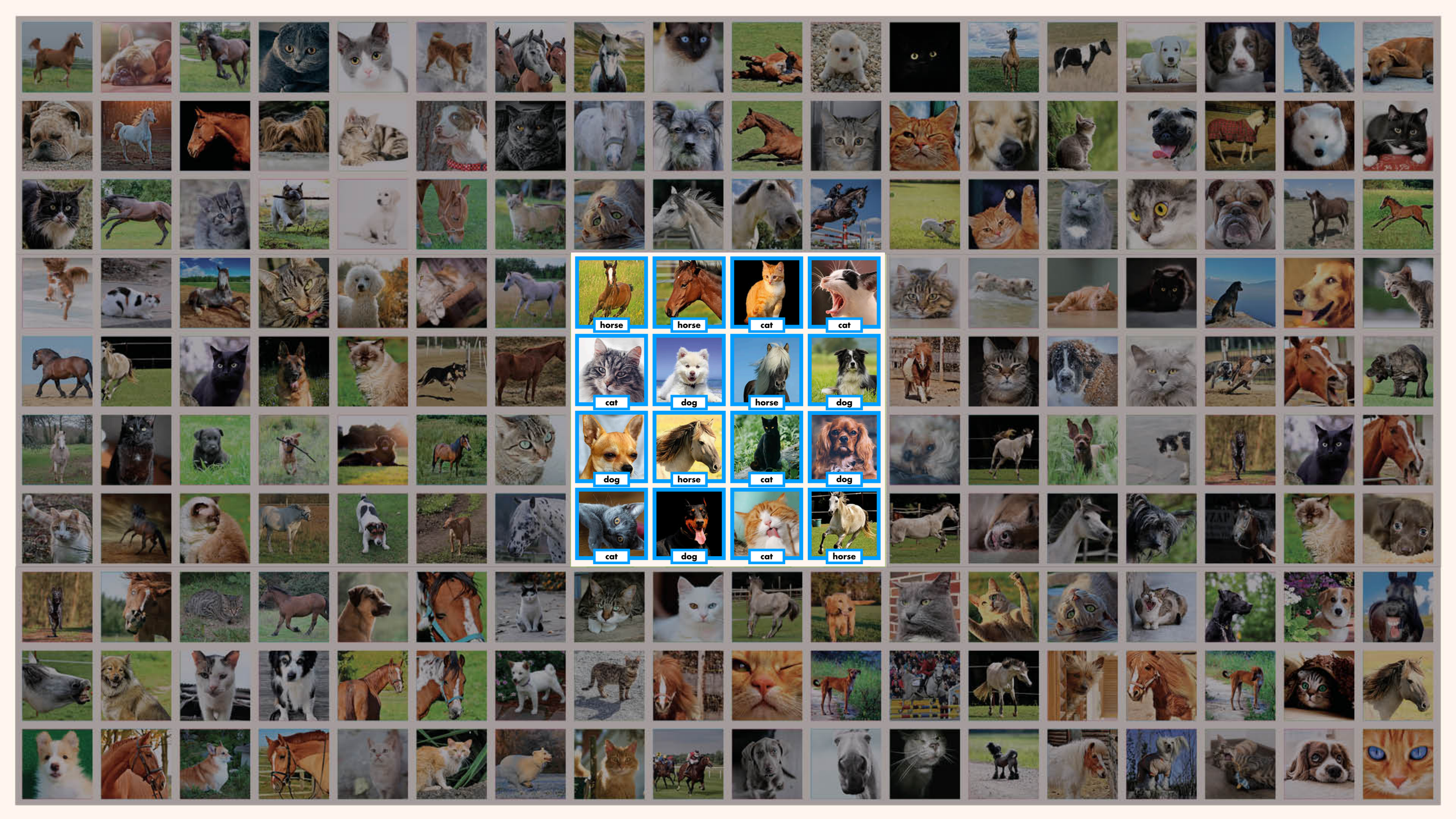
- Improves over the state-of-the-art semi-supervised techniques
  - Works much better when the number of labeled examples is scarce
- Overcomes the limitation of Temporal Ensembling
  - Fails for large number of training examples
  - Not suited for online learning
- Averaging model weights over training steps rather than final predictions
  - These weight averages improve all layer outputs, not just top output
  - Leads to better intermediate representations
  - Approach scales to large datasets and on-line learning
  - The more acccurate target labels leads to a faster feedback loop between the student and teacher models.

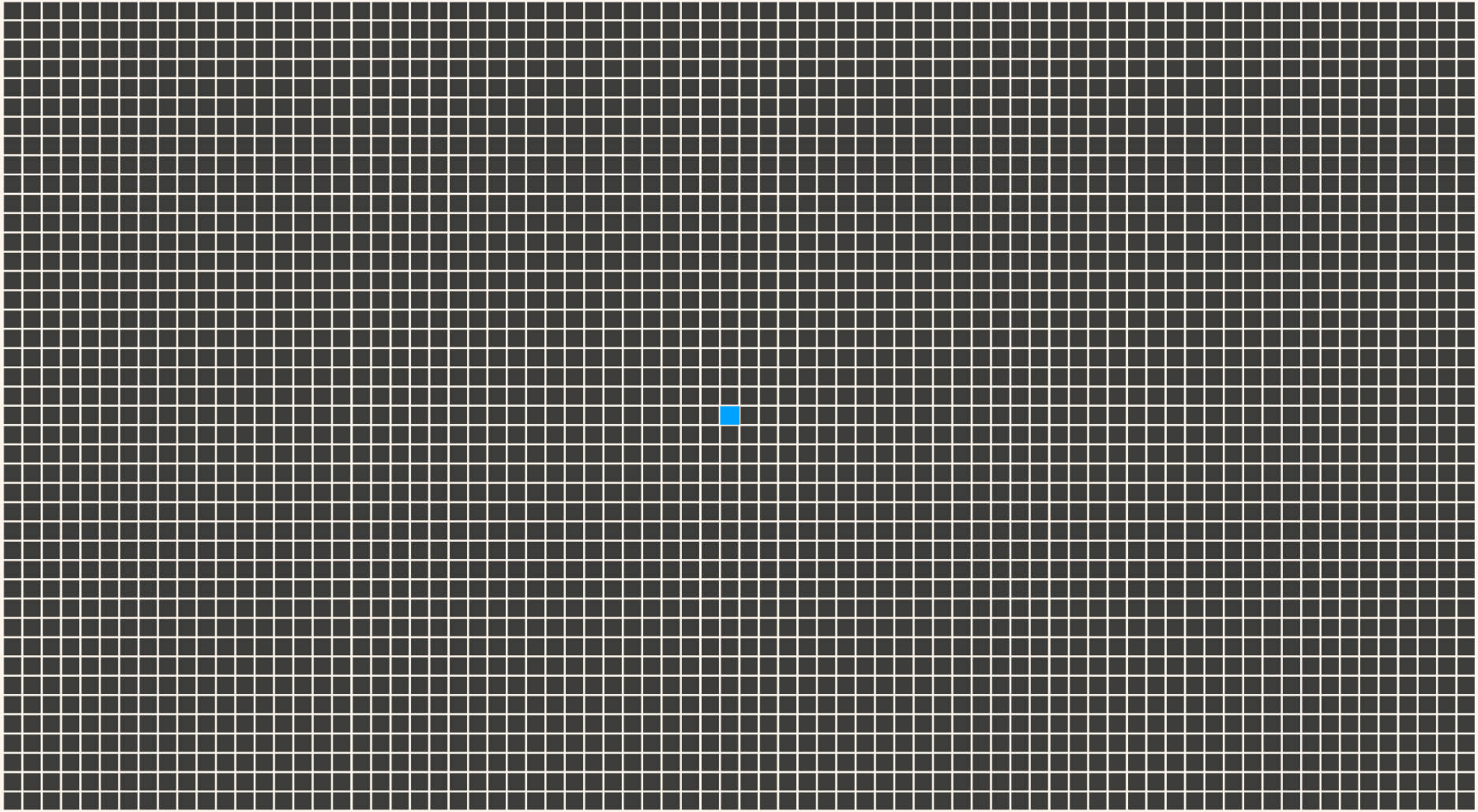
HOW THE MEAN TEACHER WORKS

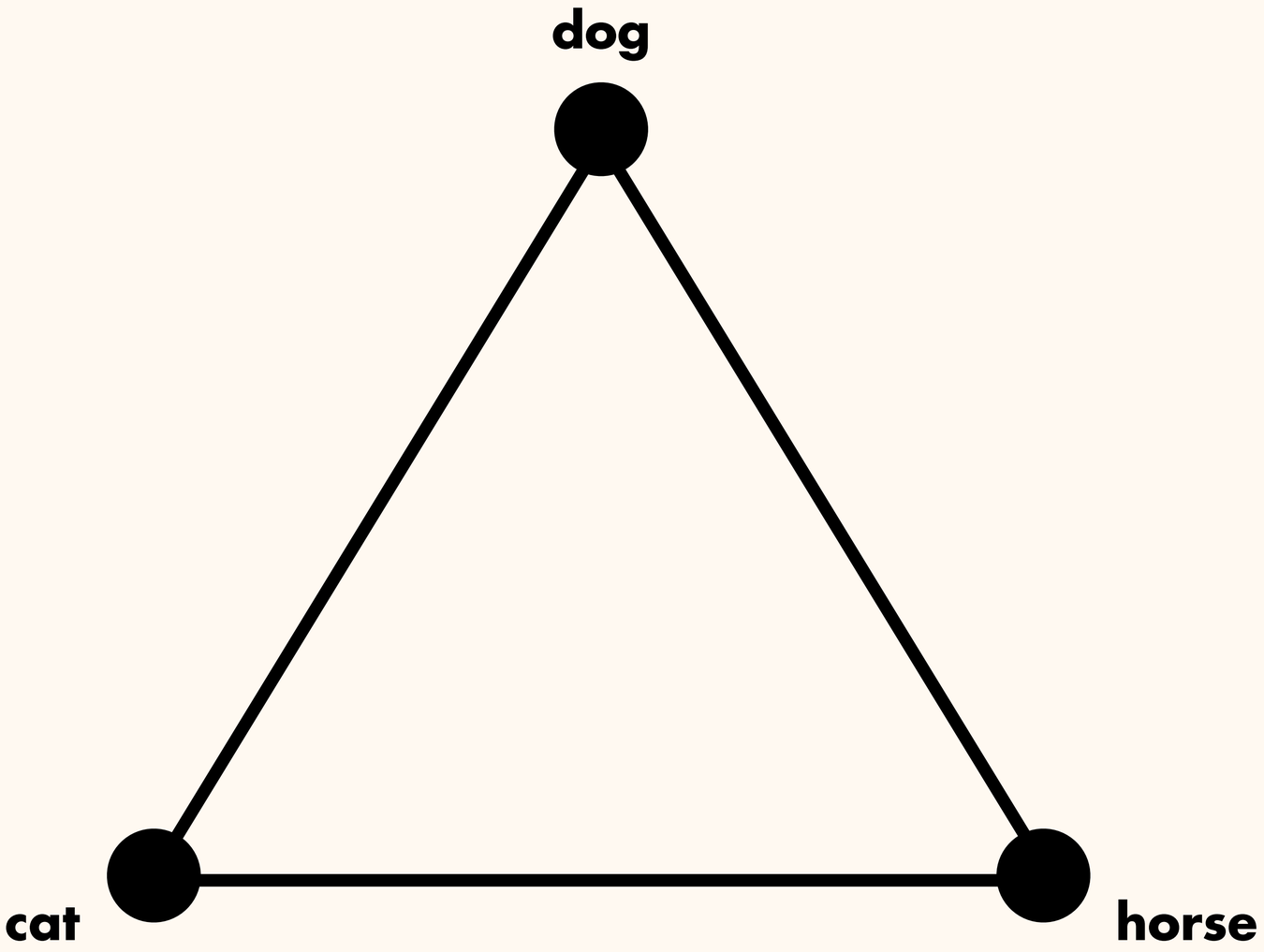
THE CURIOUS AI COMPANY

THE CURIOUS AI COMPANY

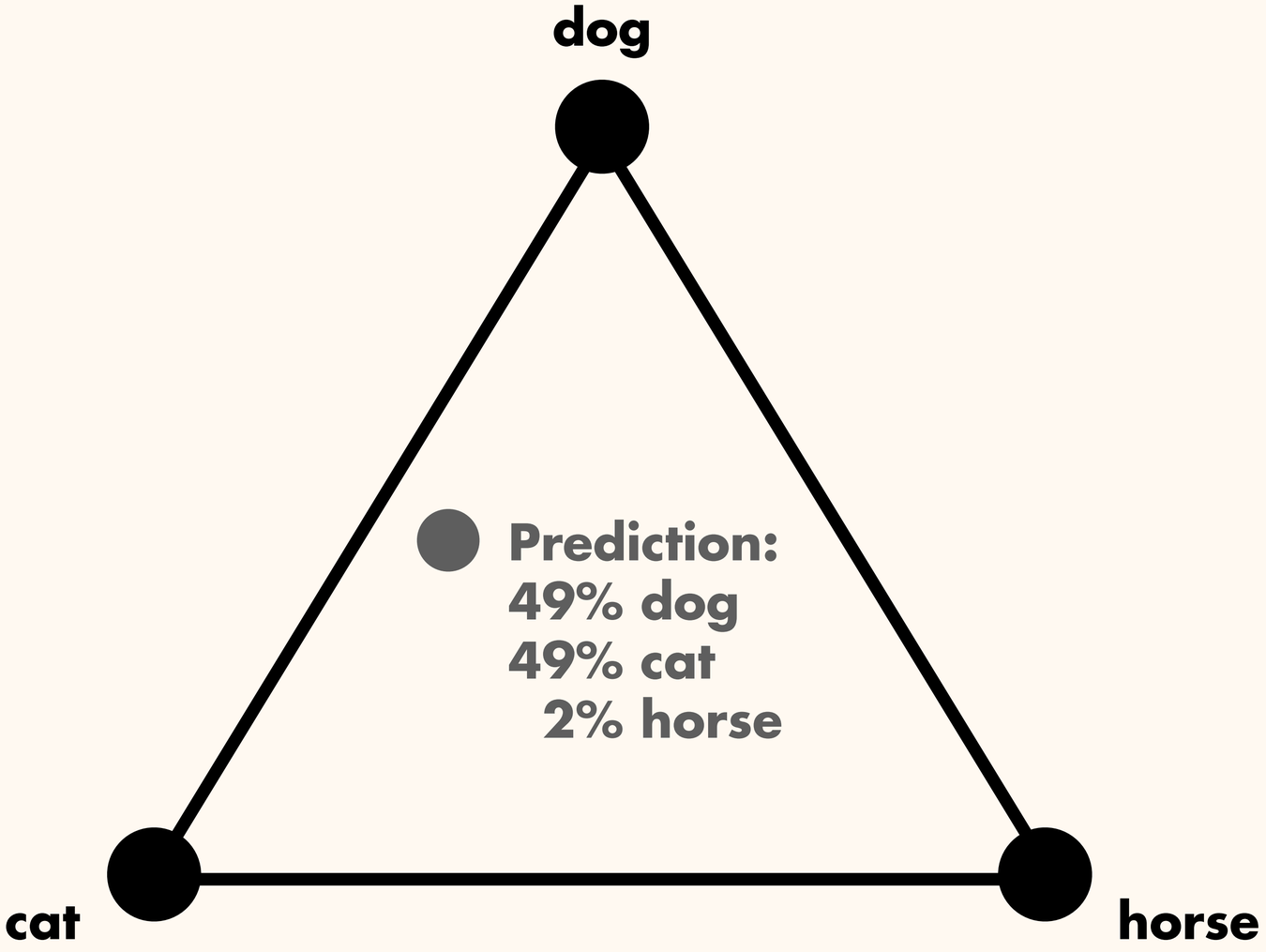
THE CURIOUS AI COMPANY


horse
horse
cat
cat
cat
dog
horse
dog
dog
horse
cat
dog
cat
dog
cat
horse

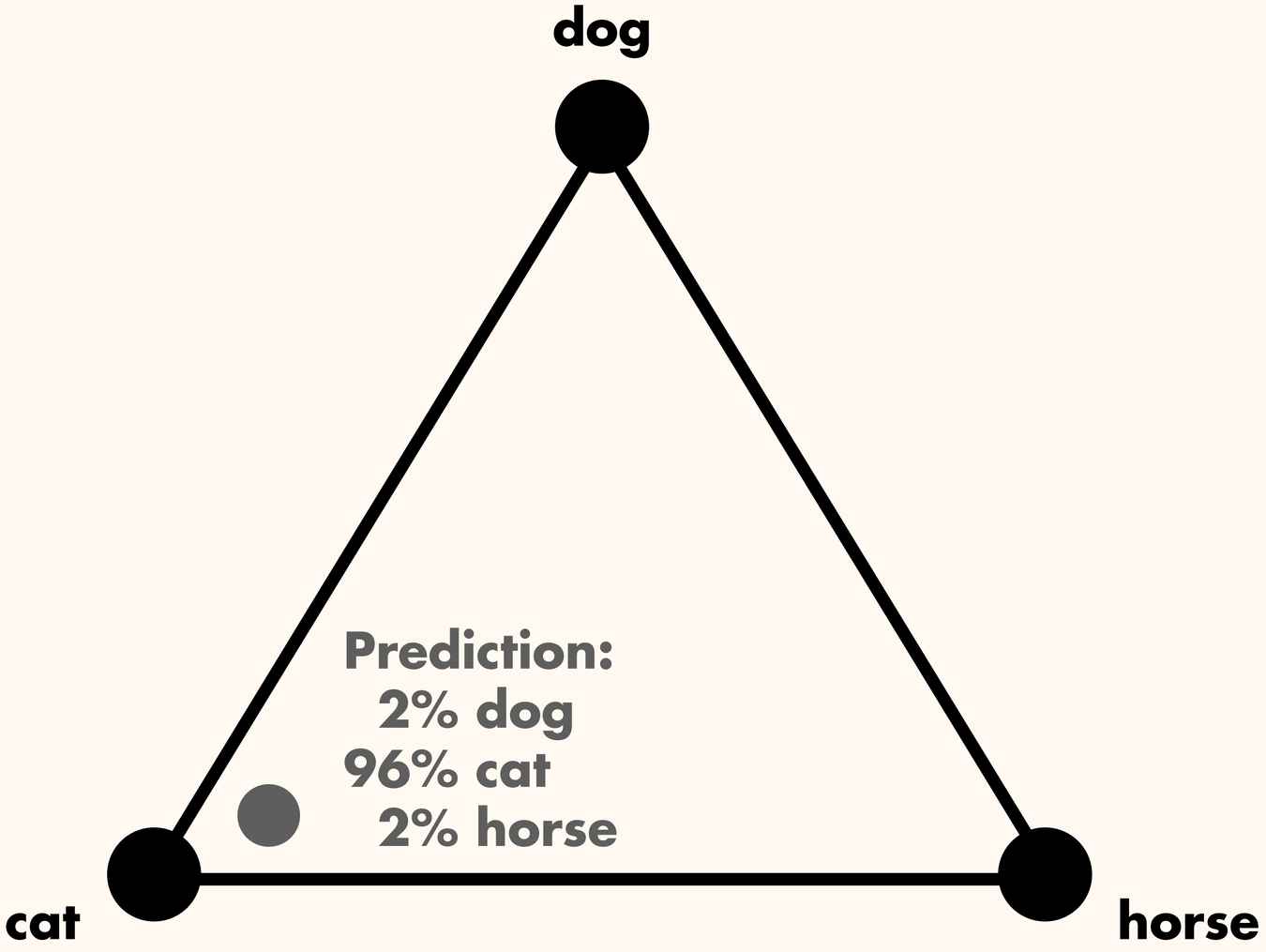


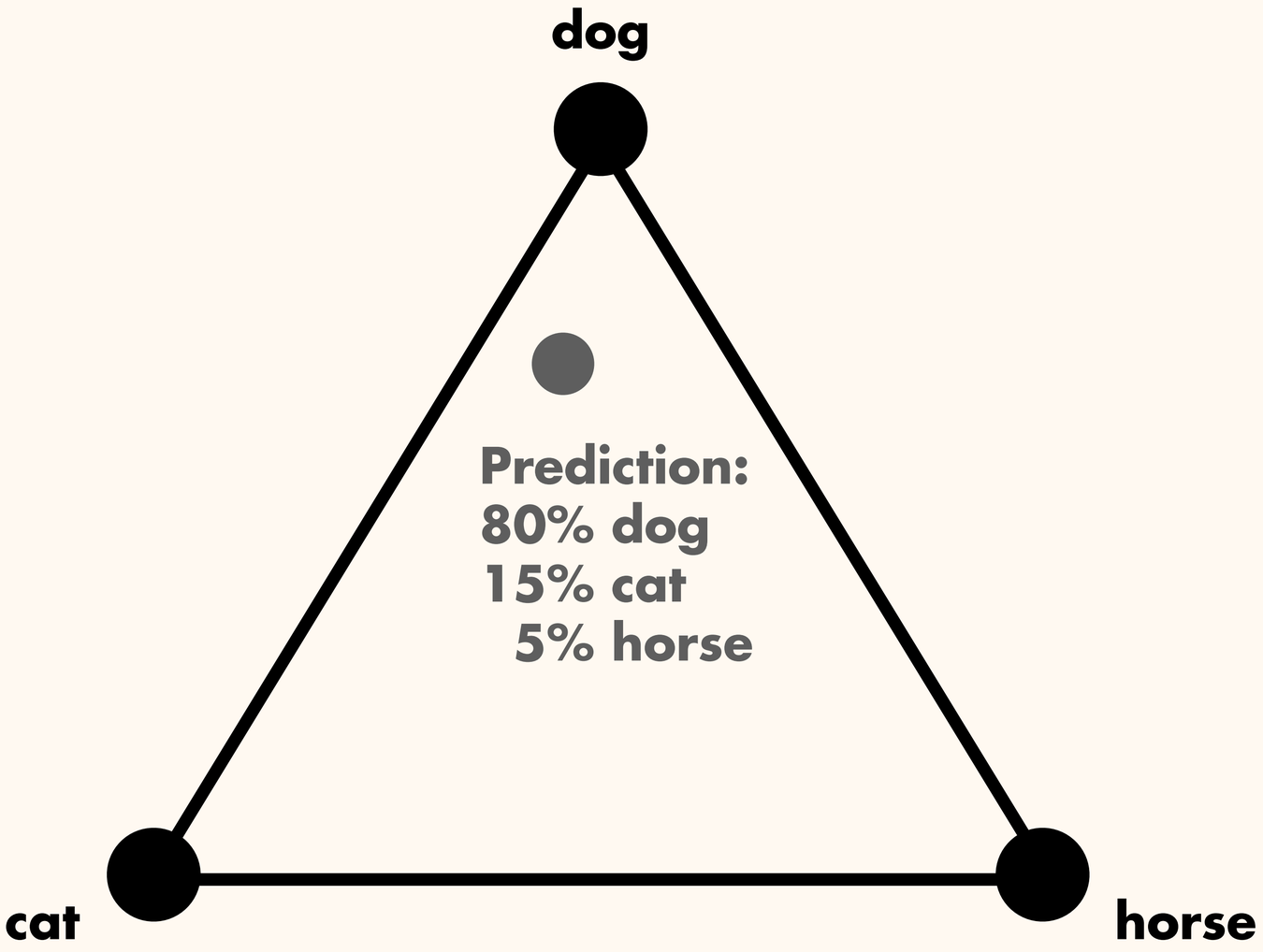


THE CURIOUS AI COMPANY



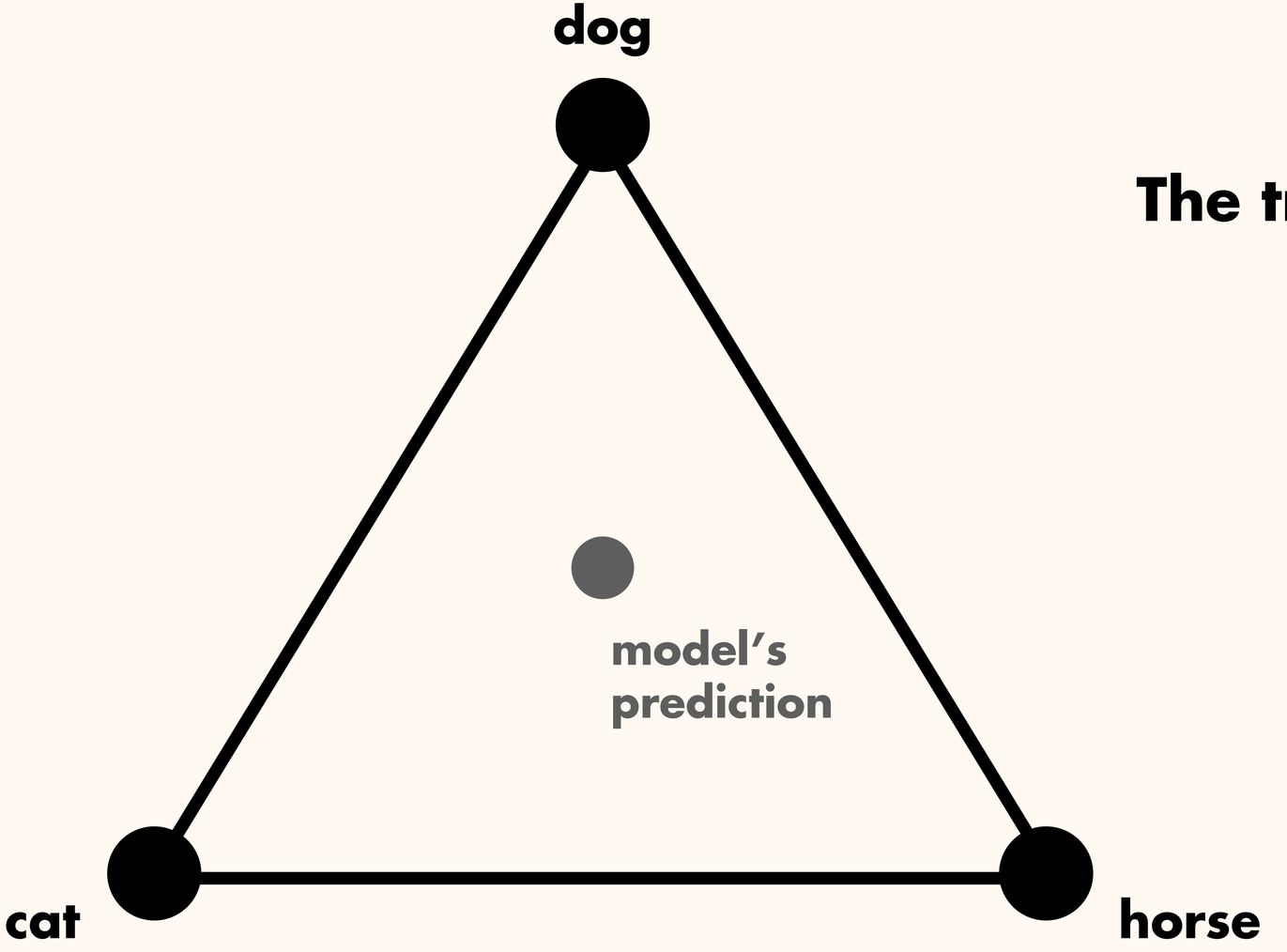
THE CURIOUS AI COMPANY





THE CURIOUS AI COMPANY

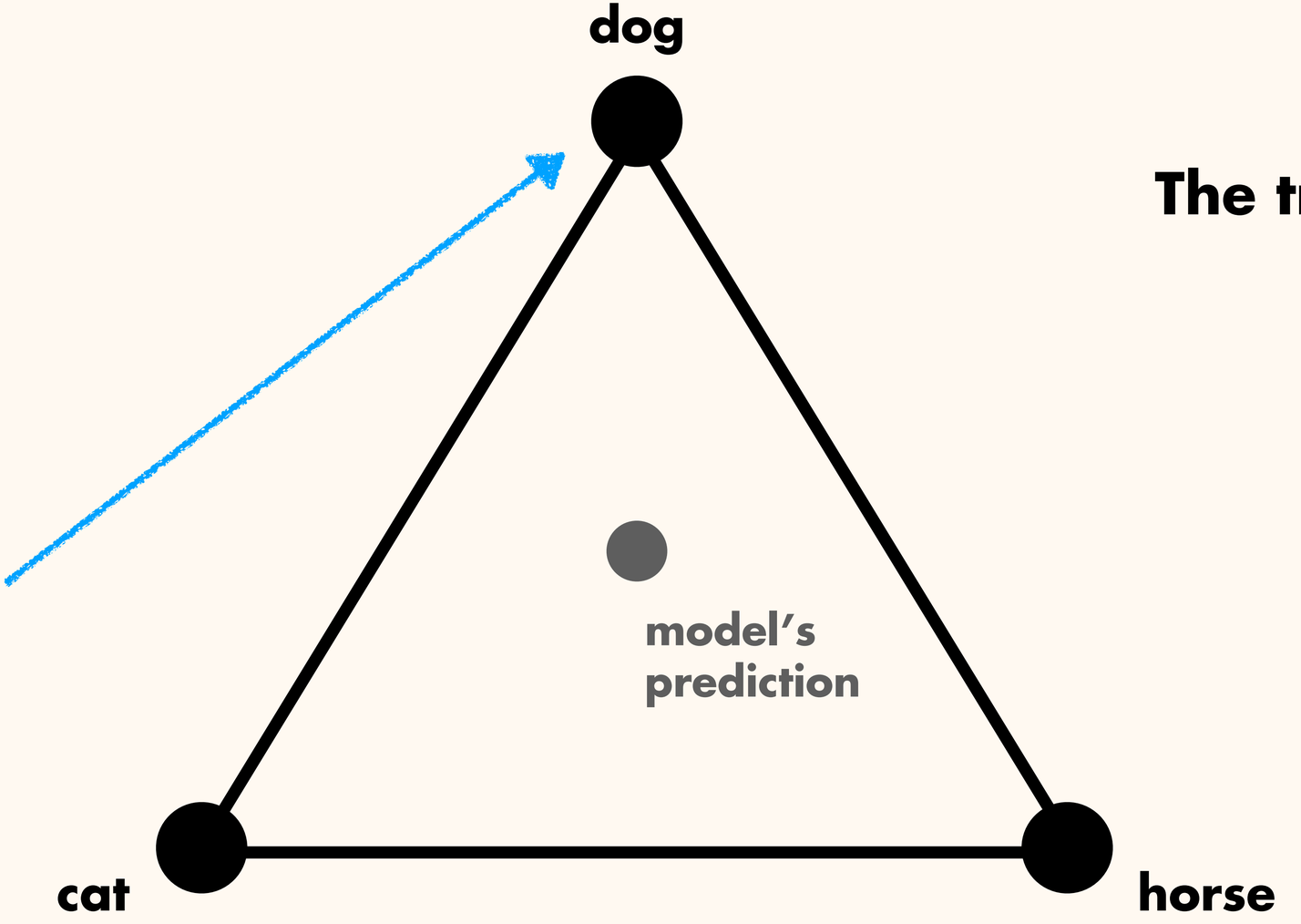
SUPERVISED LEARNING



The true label pulls predictions to its direction.

THE CURIOUS AI COMPANY

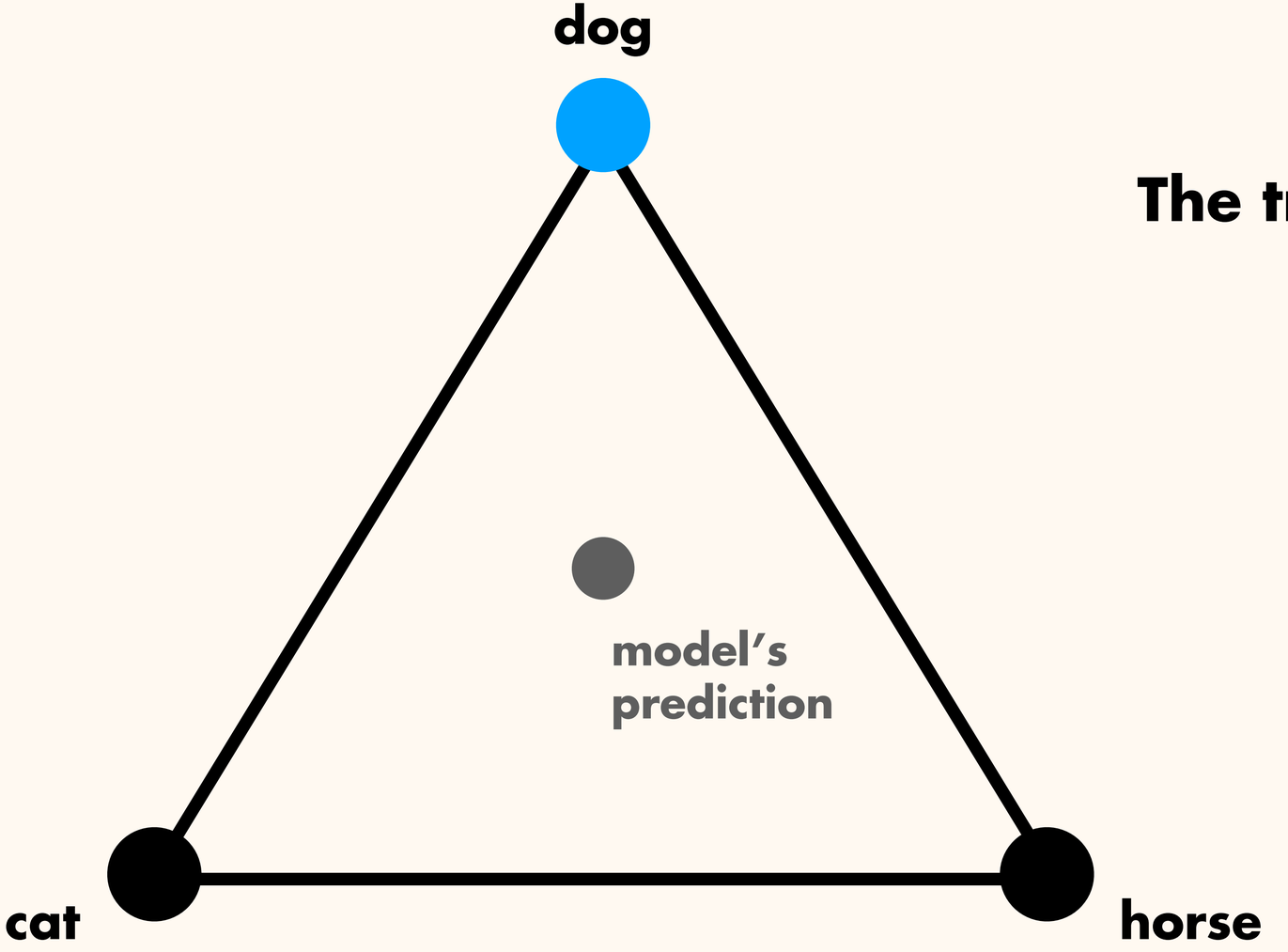
SUPERVISED LEARNING



The true label pulls predictions to its direction.

THE CURIOUS AI COMPANY

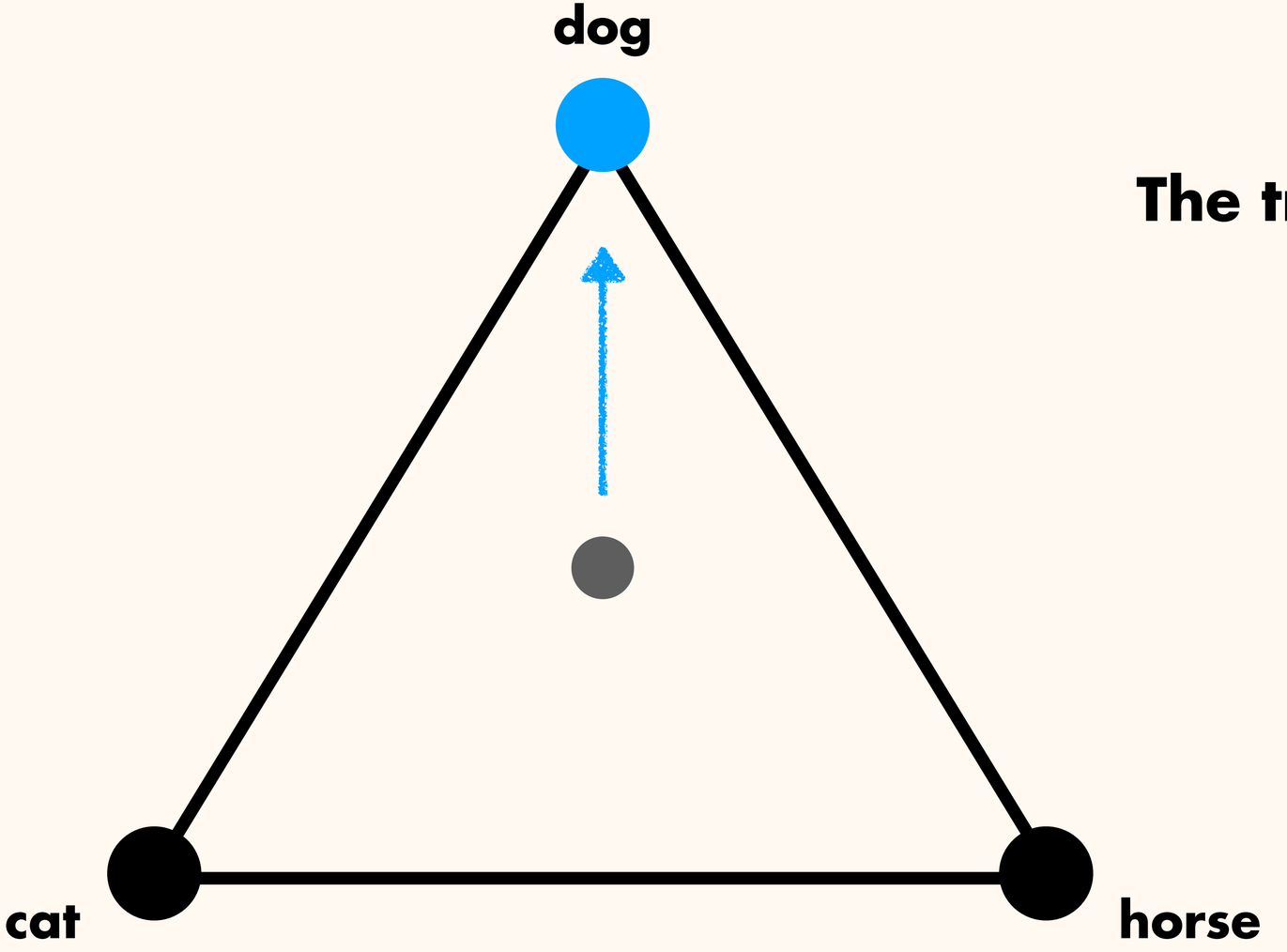
SUPERVISED LEARNING



The true label pulls predictions to its direction.

THE CURIOUS AI COMPANY

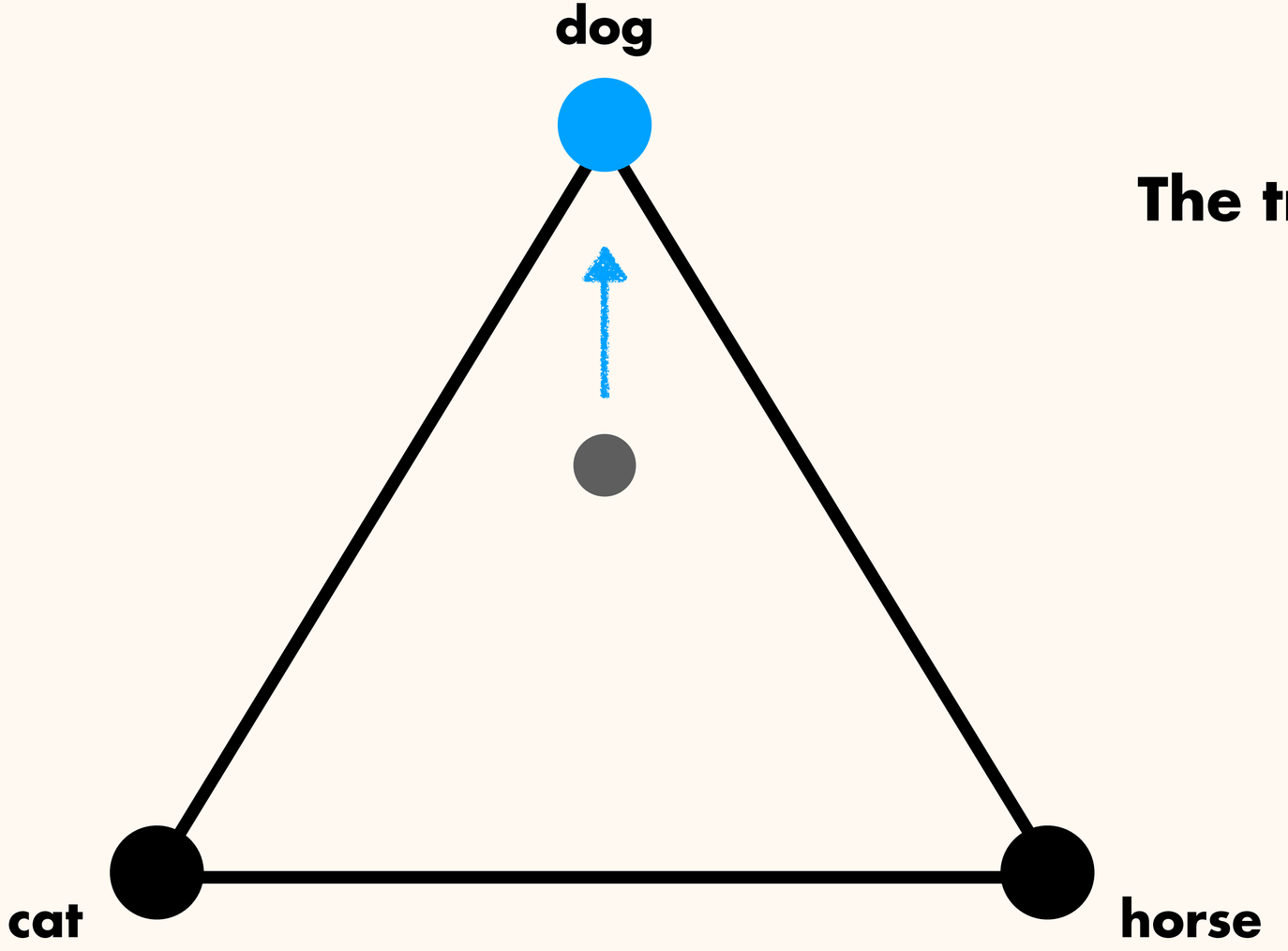
SUPERVISED LEARNING



The true label pulls predictions to its direction.

THE CURIOUS AI COMPANY

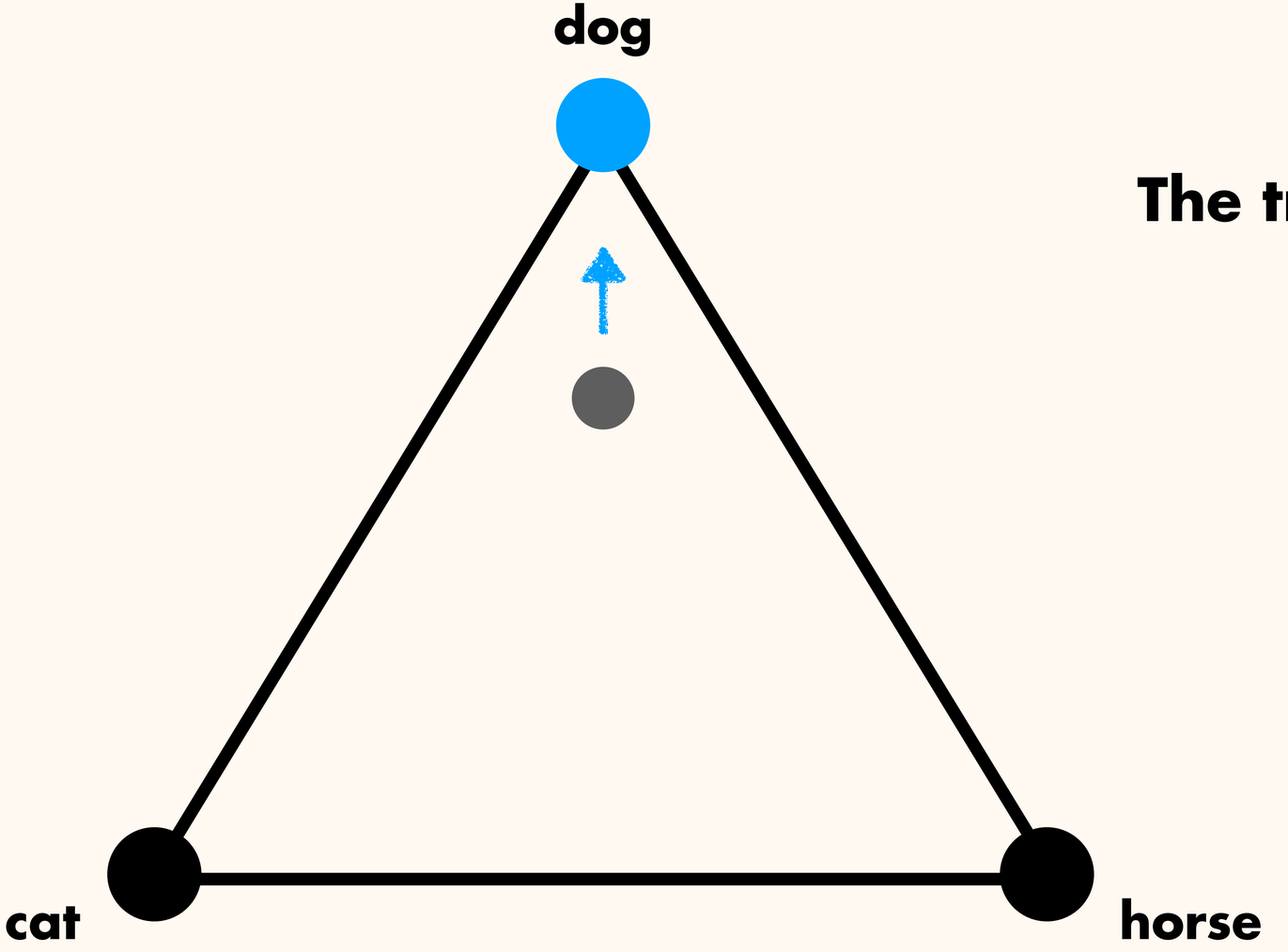
SUPERVISED LEARNING



The true label pulls predictions to its direction.

THE CURIOUS AI COMPANY

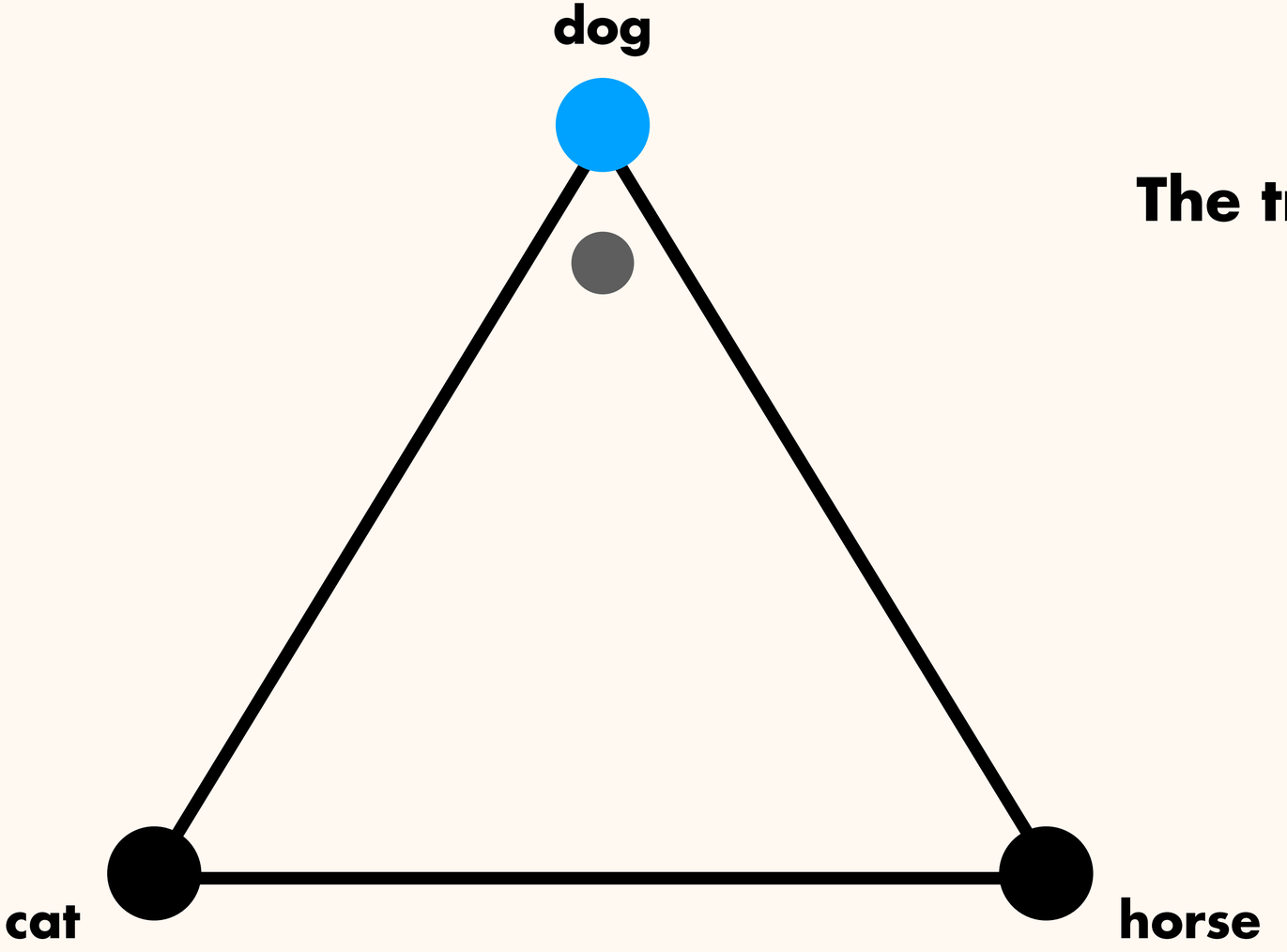
SUPERVISED LEARNING



The true label pulls predictions to its direction.

THE CURIOUS AI COMPANY

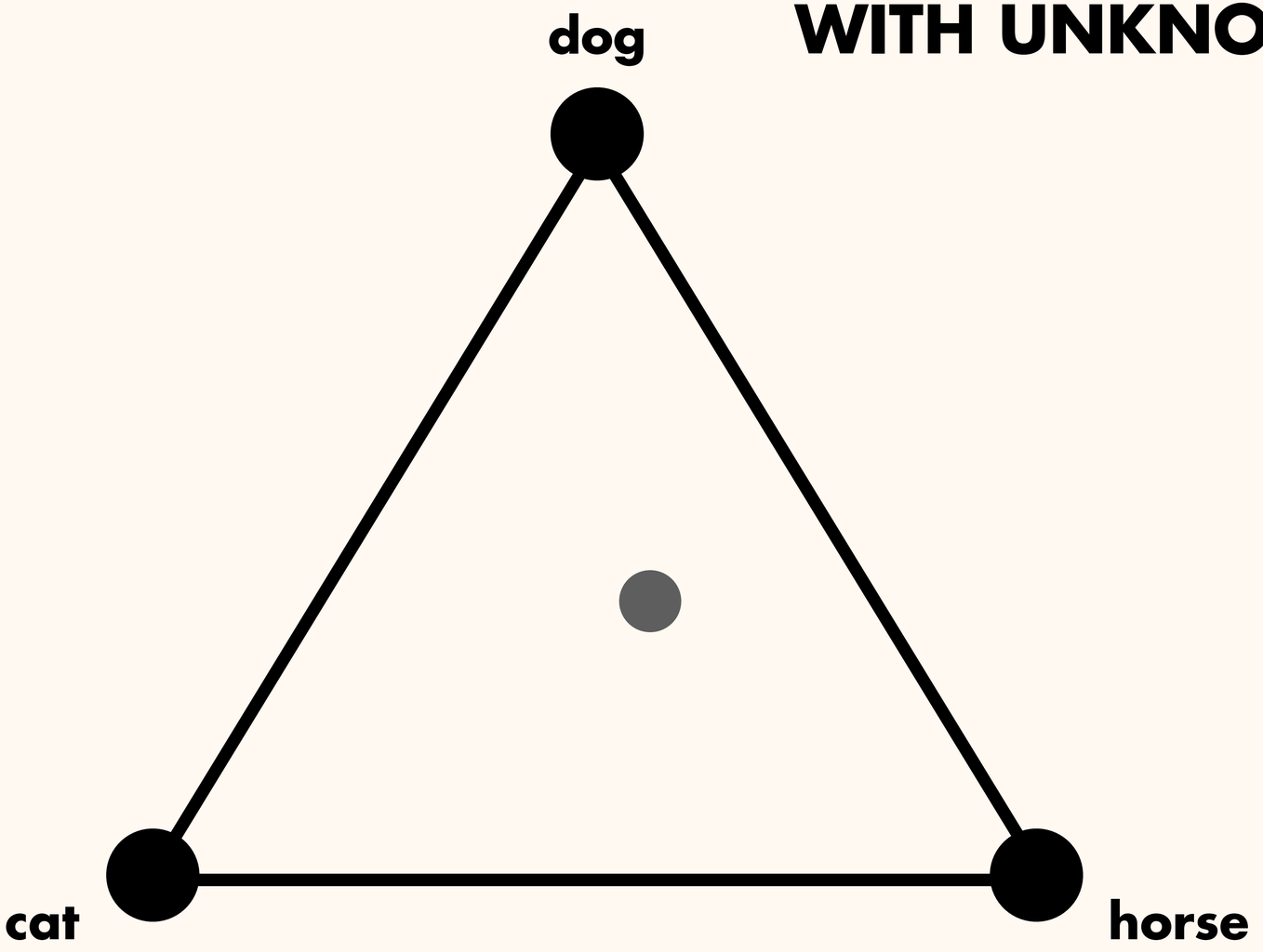
SUPERVISED LEARNING



The true label pulls predictions to its direction.

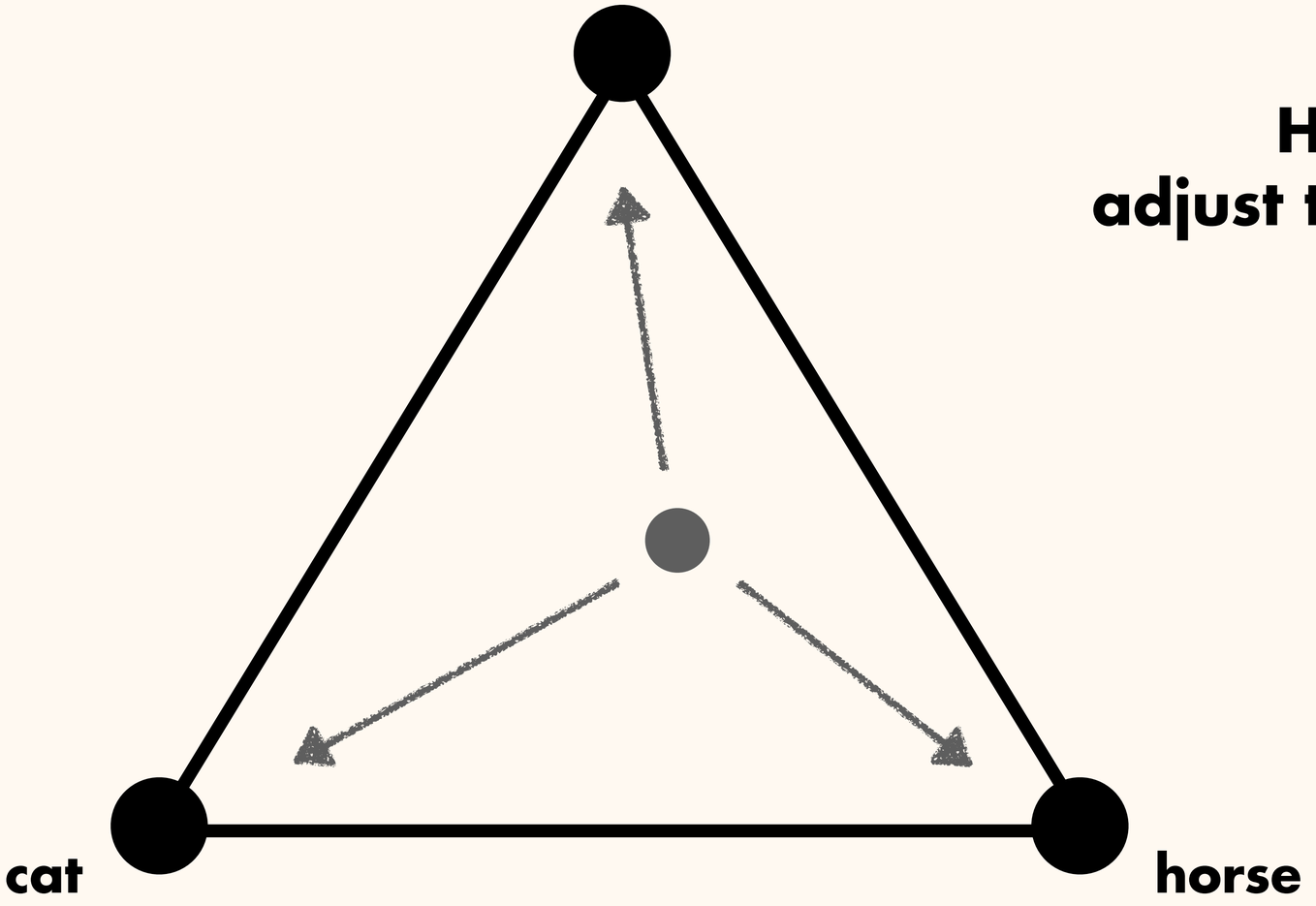
THE CURIOUS AI COMPANY

WHAT ABOUT EXAMPLES WITH UNKNOWN LABELS?



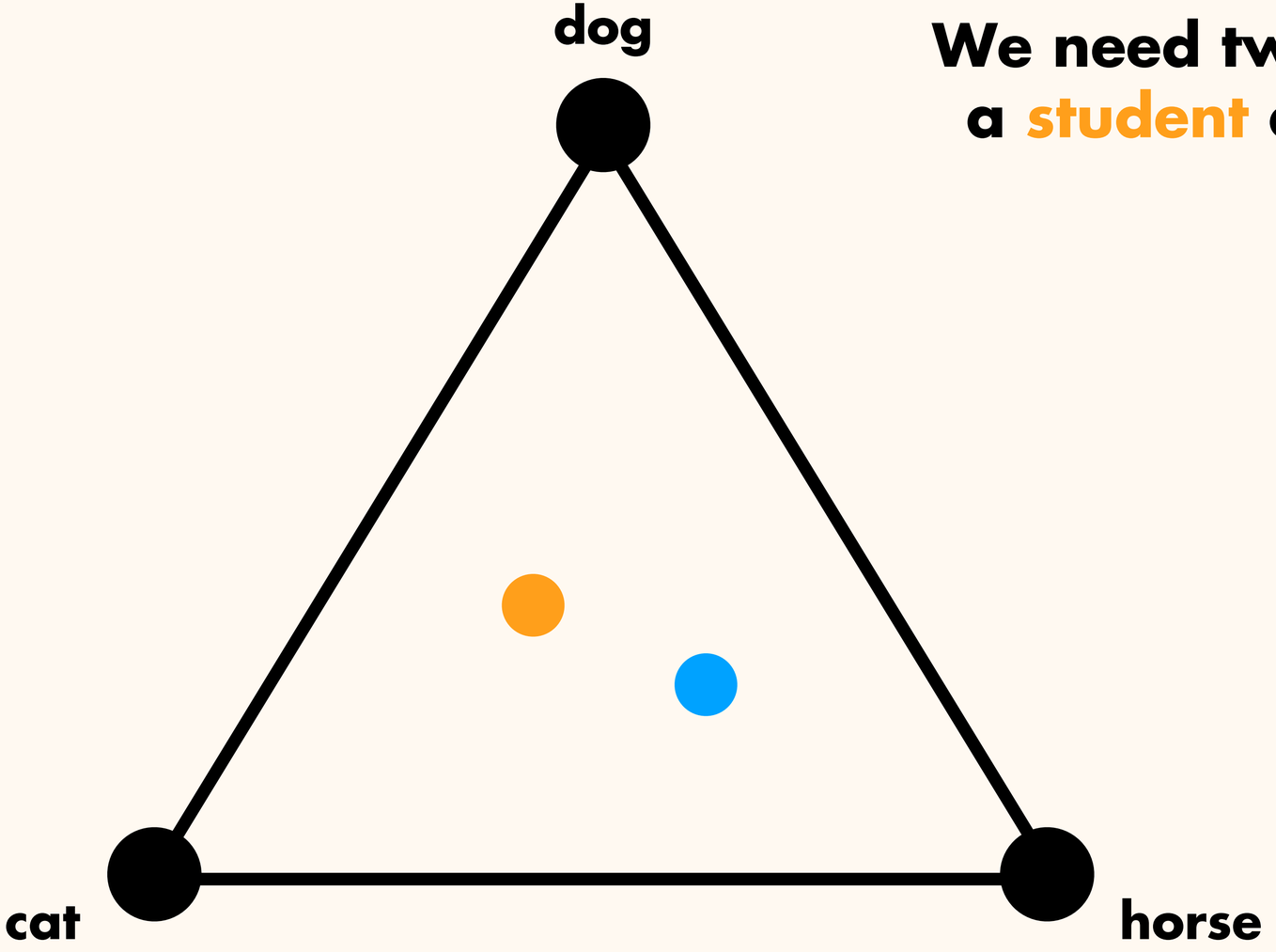
THE CURIOUS AI COMPANY

WHAT ABOUT EXAMPLES WITH UNKNOWN LABELS?



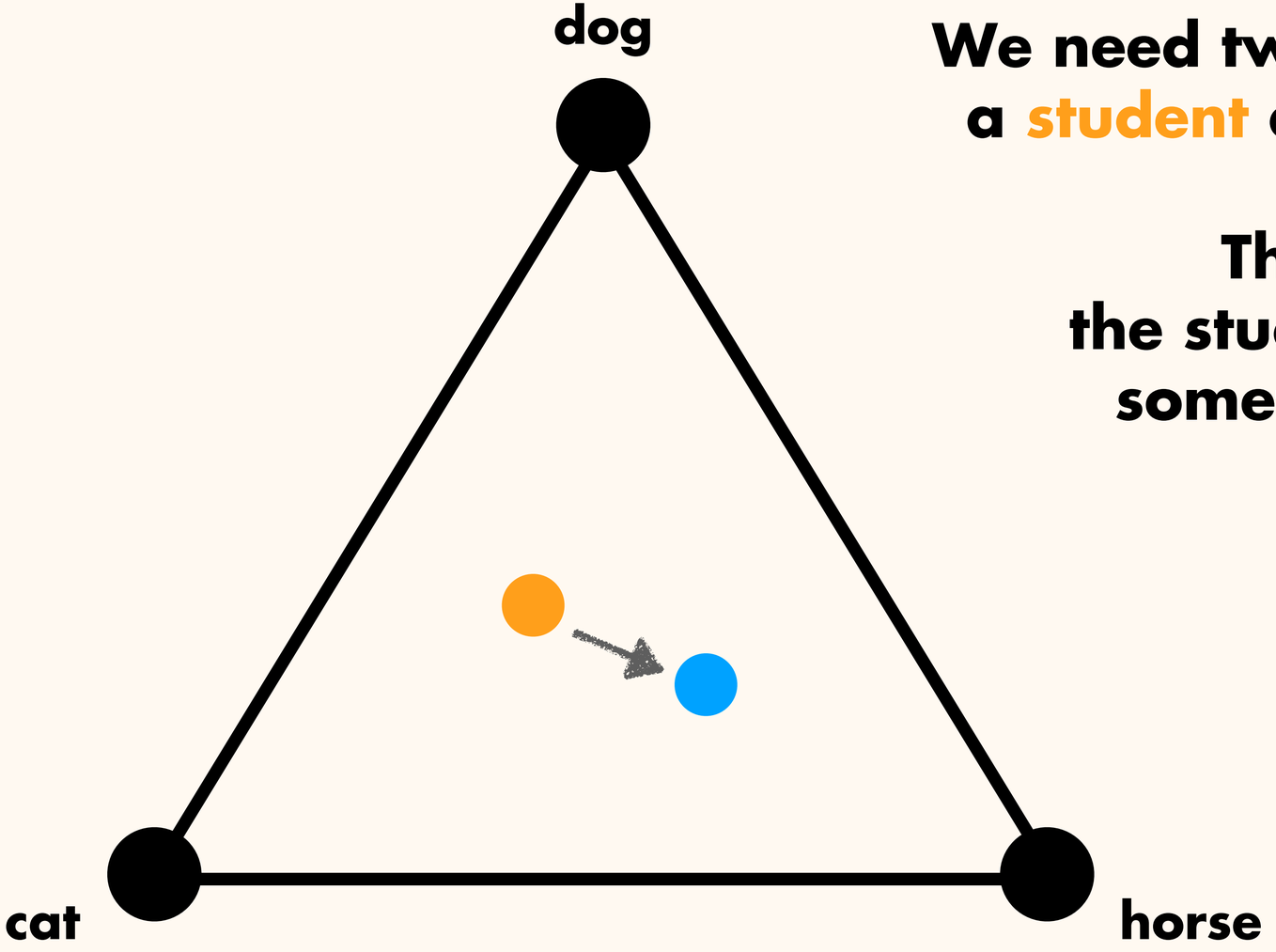
How should we adjust the prediction?

THE CURIOUS AI COMPANY



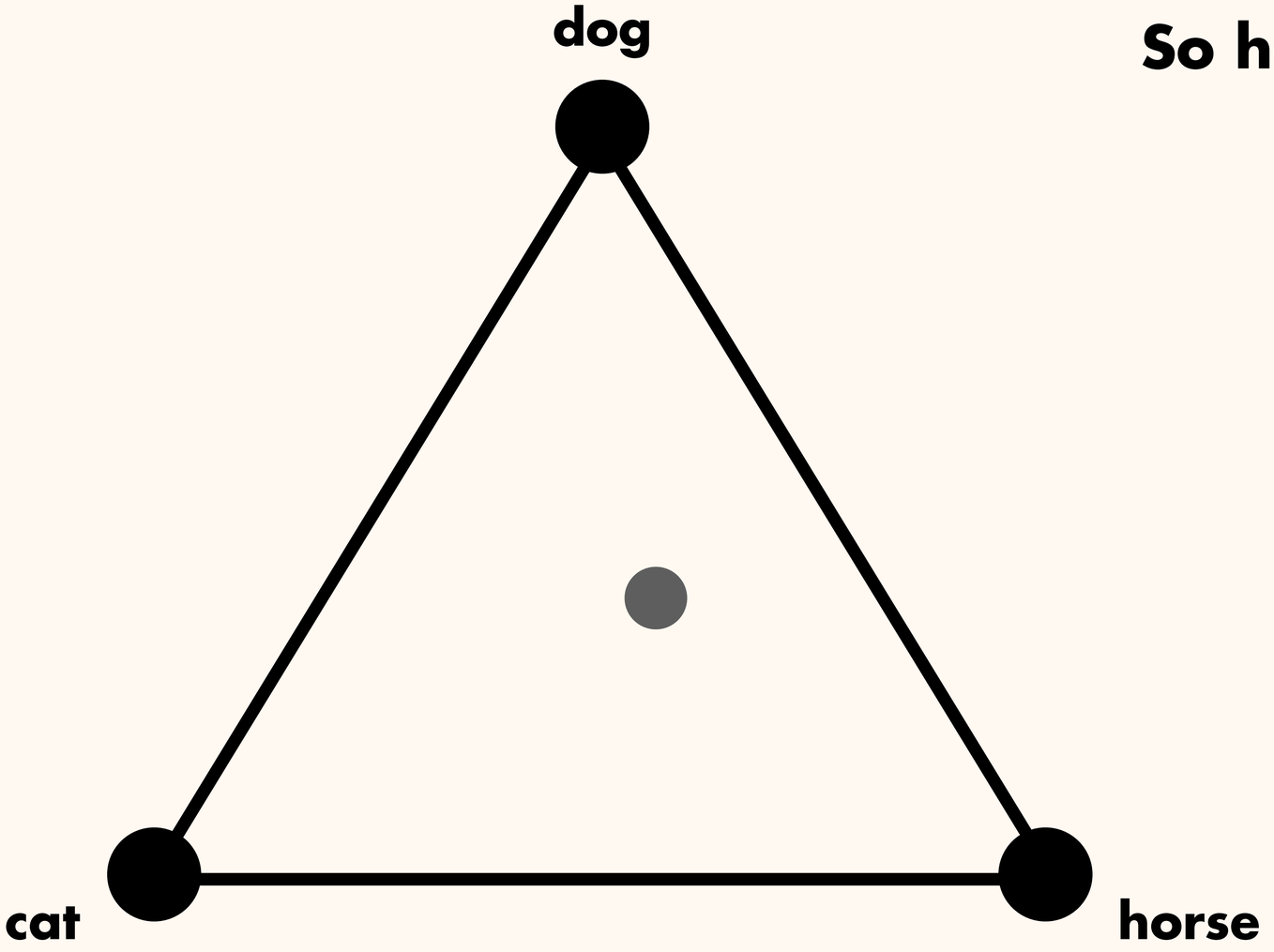
We need two predictions:
a student and a teacher.

THE CURIOUS AI COMPANY



We need two predictions:
a student and a teacher.

Then, hopefully,
the student can learn
something from the
teacher.

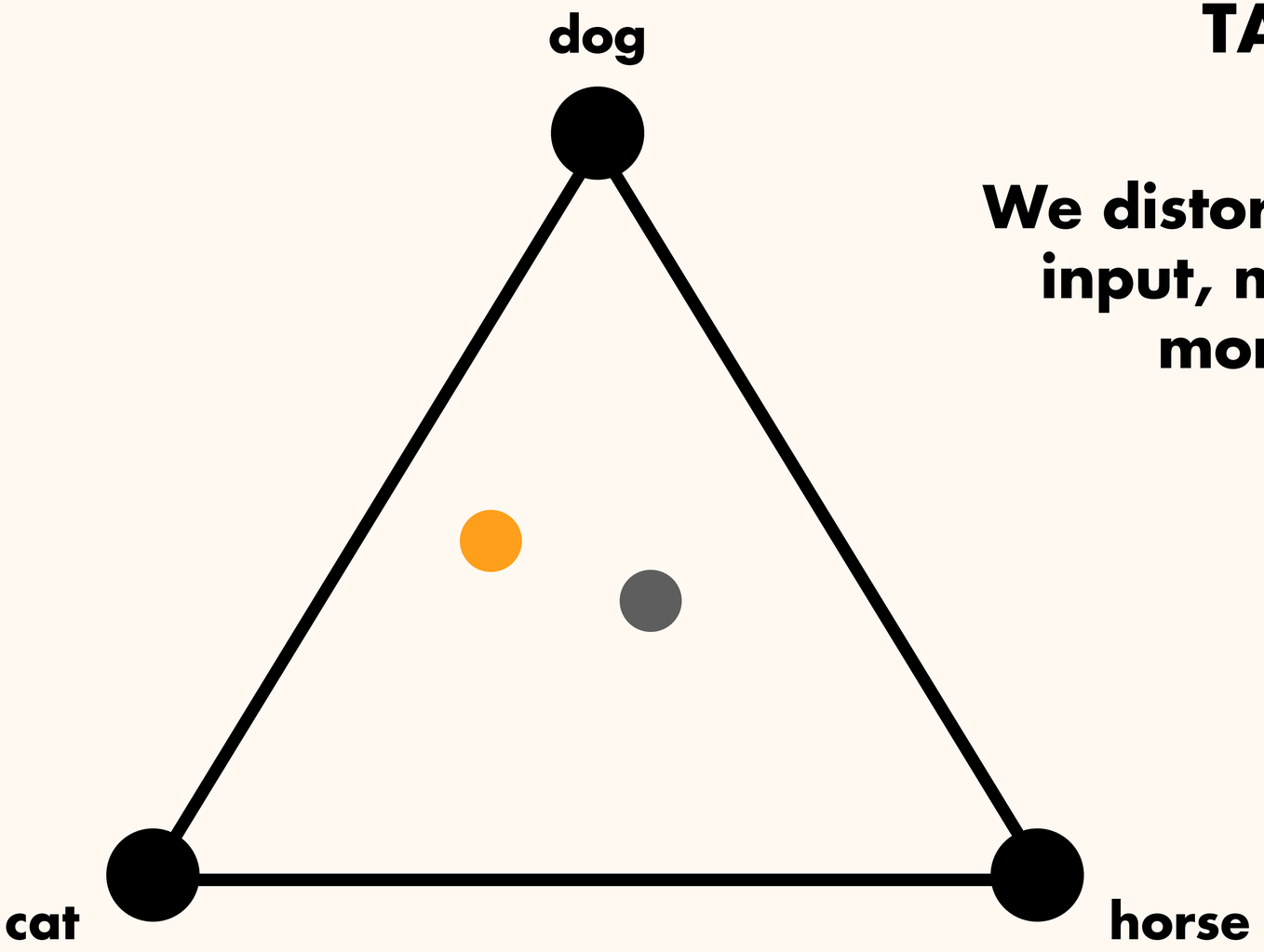


So how to do this?

Two ways.

THE CURIOUS AI COMPANY

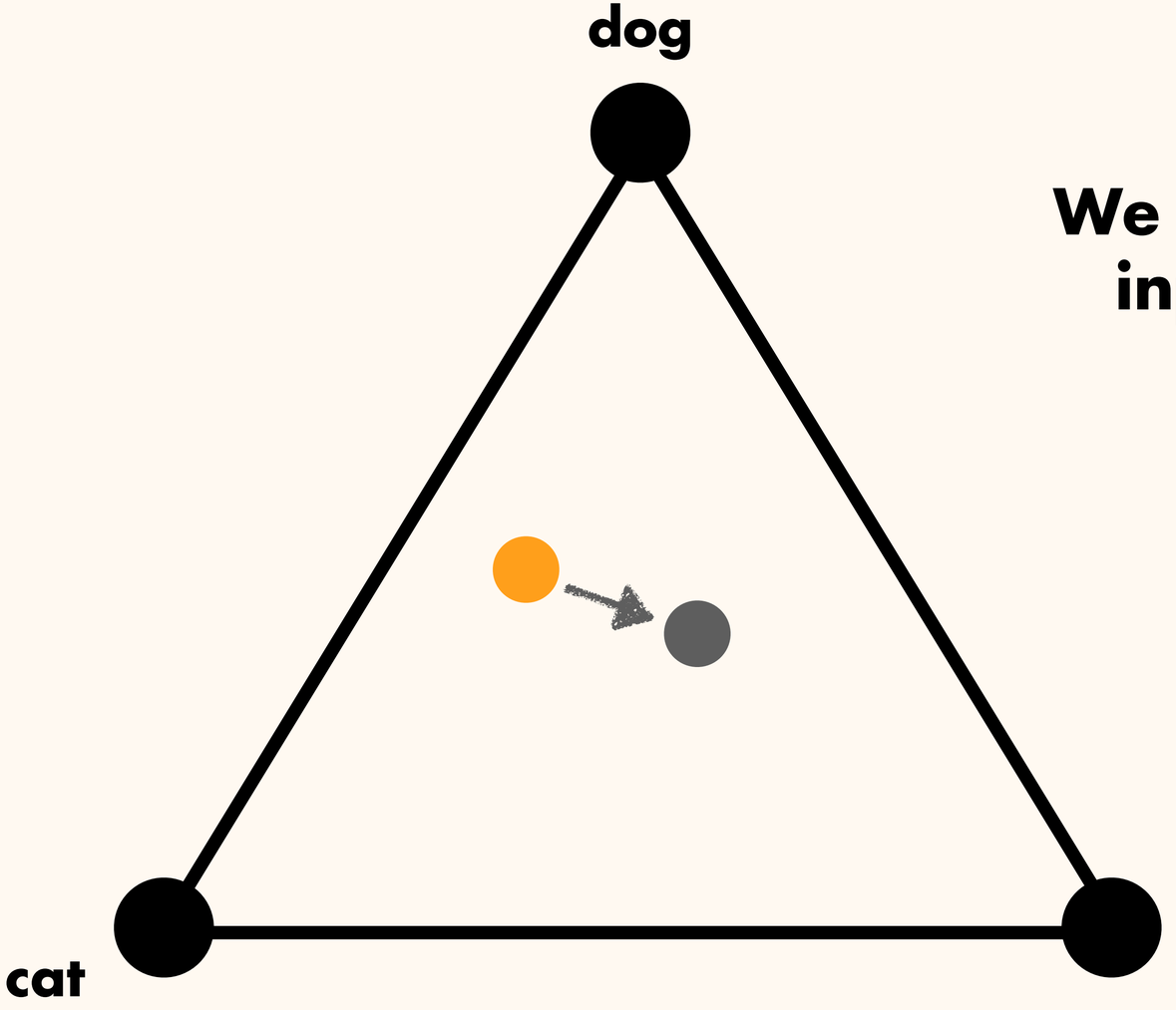
1. MAKE THE TASK HARDER.



We distort the student's input, making its task more challenging.

THE CURIOUS AI COMPANY

1. MAKE THE TASK HARDER.

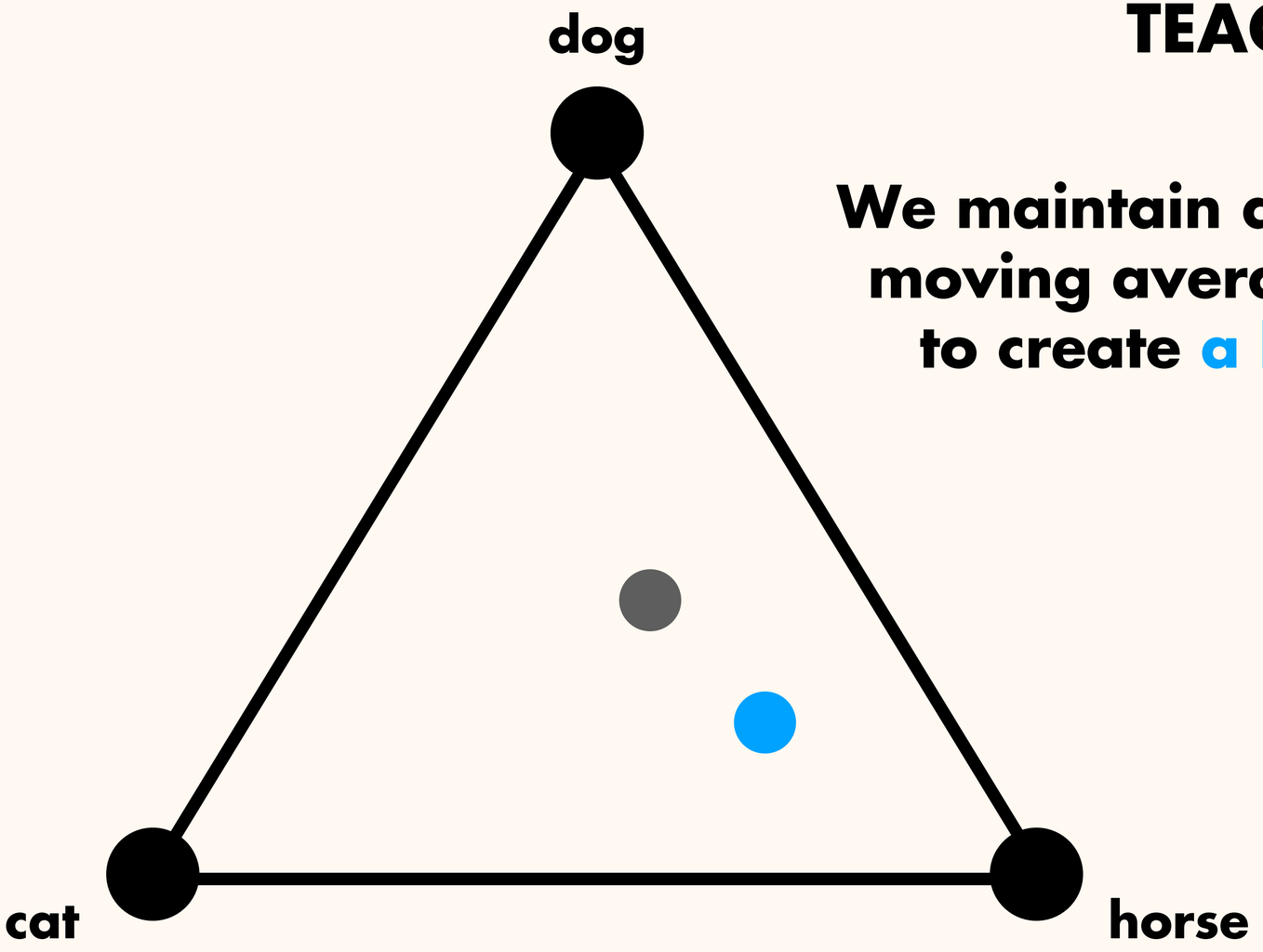


We distort the student's input, making its task more challenging.

And then we train the harder task to predict the easier tasks' output.

THE CURIOUS AI COMPANY

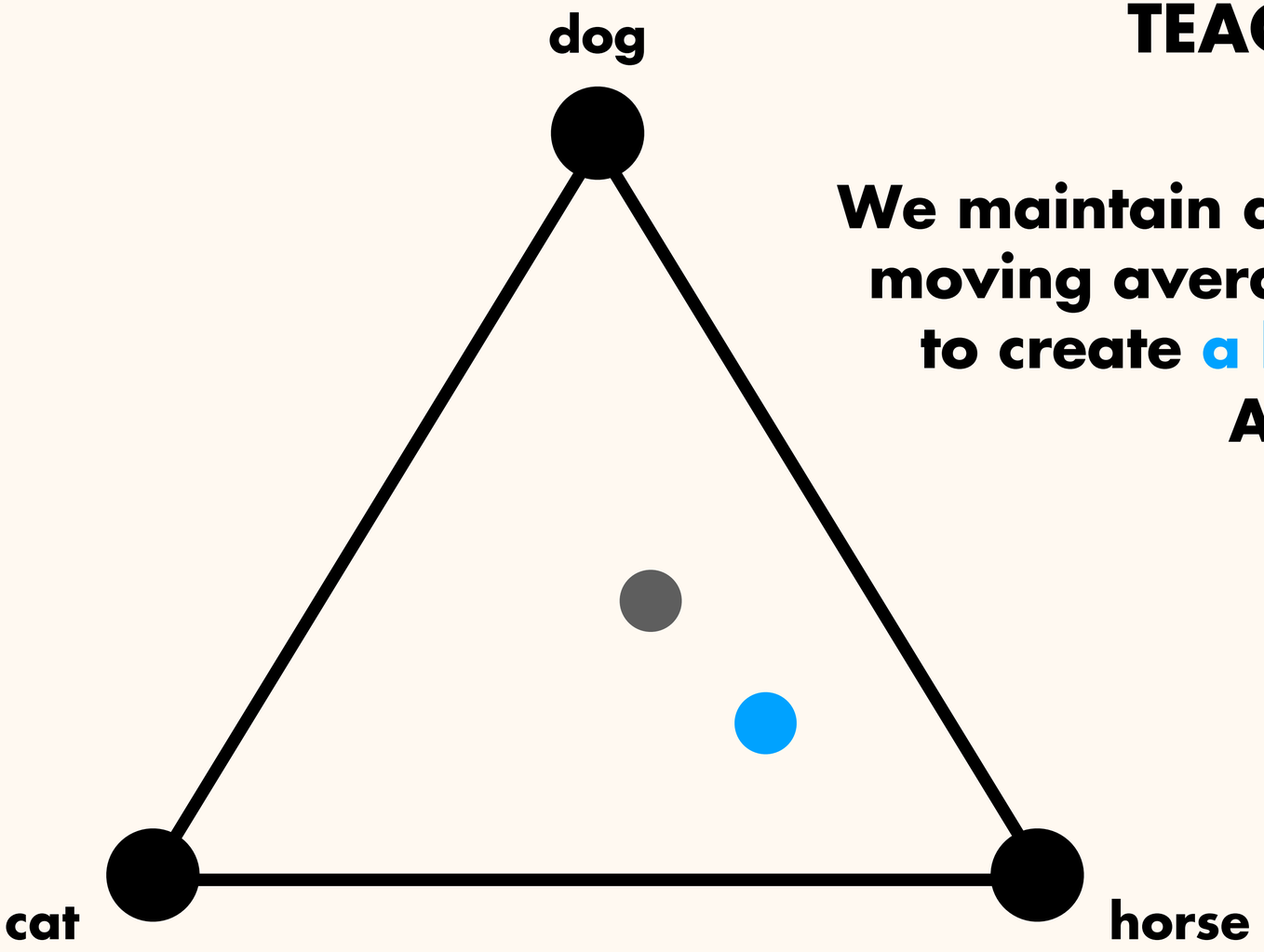
2. MAKE THE TEACHER BETTER.



We maintain an exponential moving average of weights to create a better teacher.

THE CURIOUS AI COMPANY

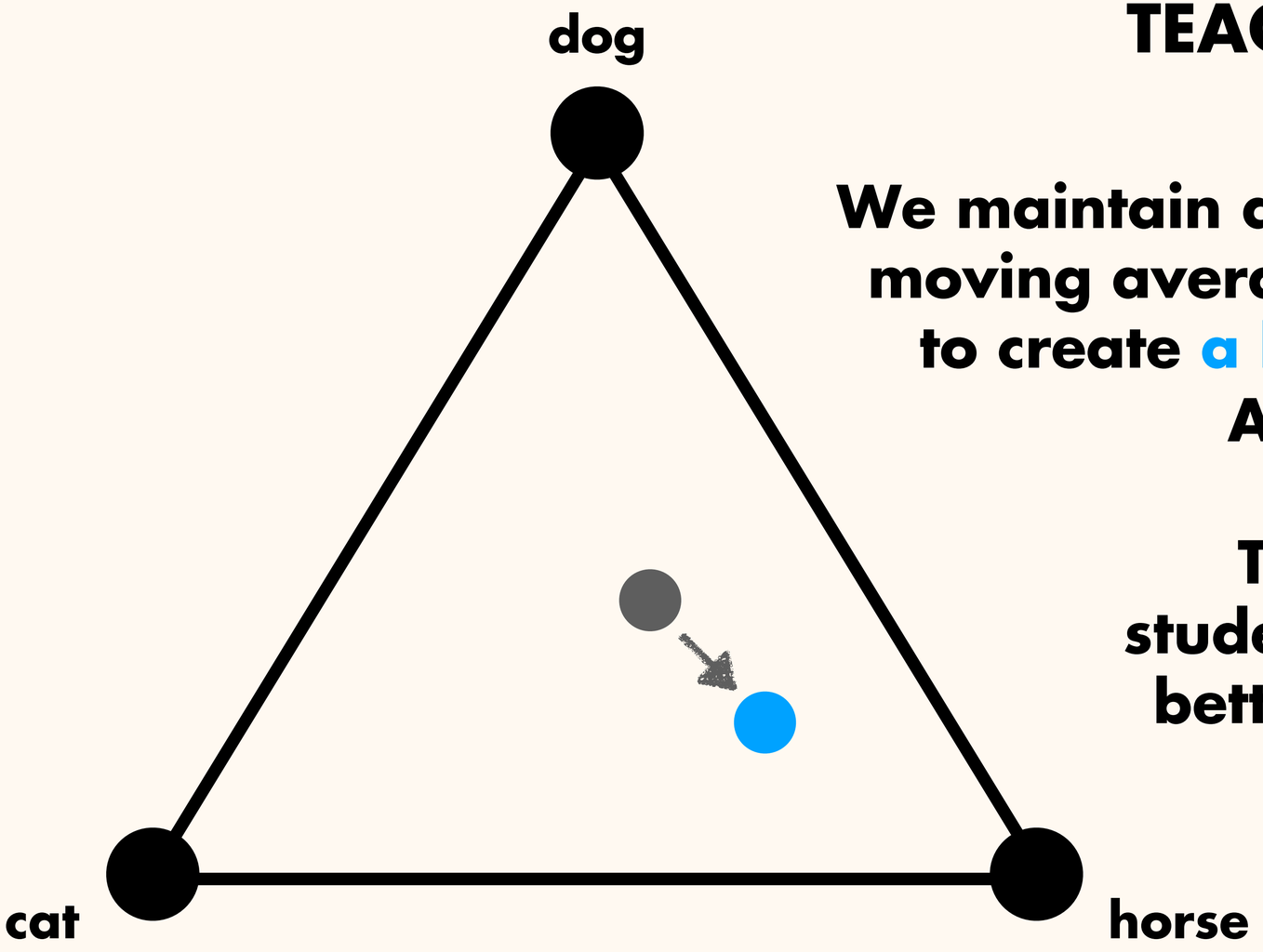
2. MAKE THE TEACHER BETTER.



We maintain an exponential moving average of weights to create a better teacher.
A *mean* teacher.

THE CURIOUS AI COMPANY

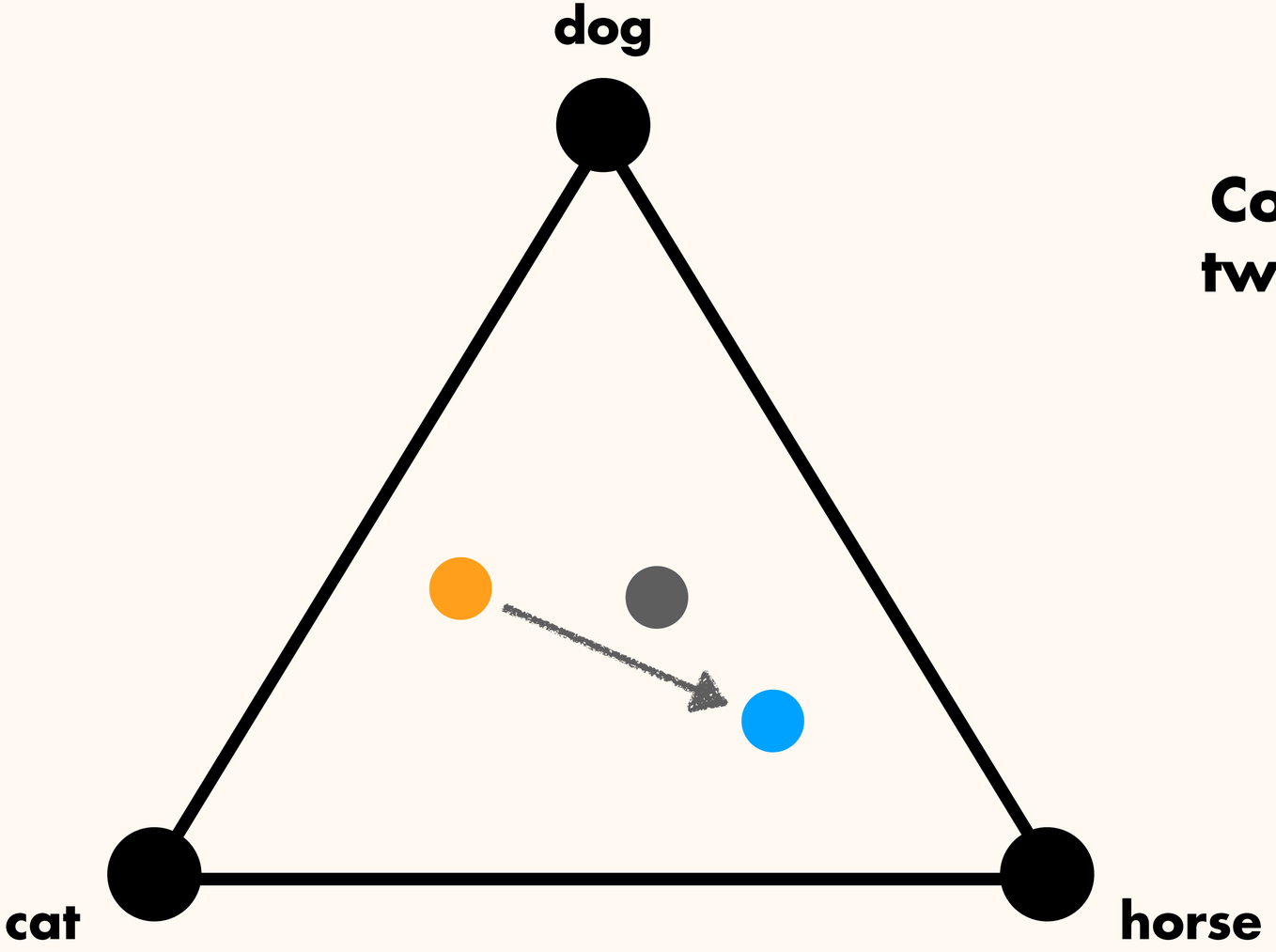
2. MAKE THE TEACHER BETTER.



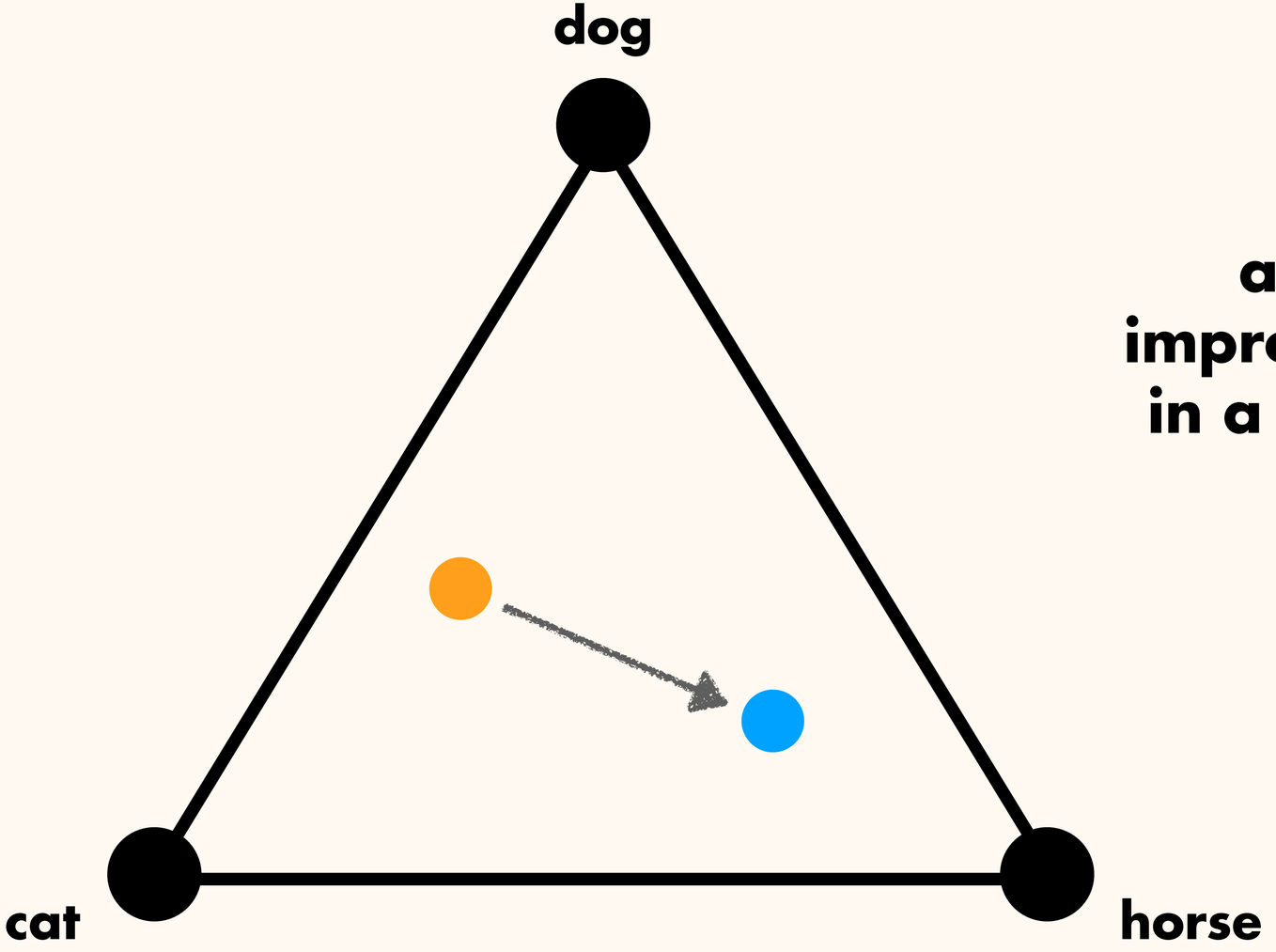
We maintain an exponential moving average of weights to create a better teacher.
A *mean* teacher.

Then we let the student learn these better predictions.

THE CURIOUS AI COMPANY

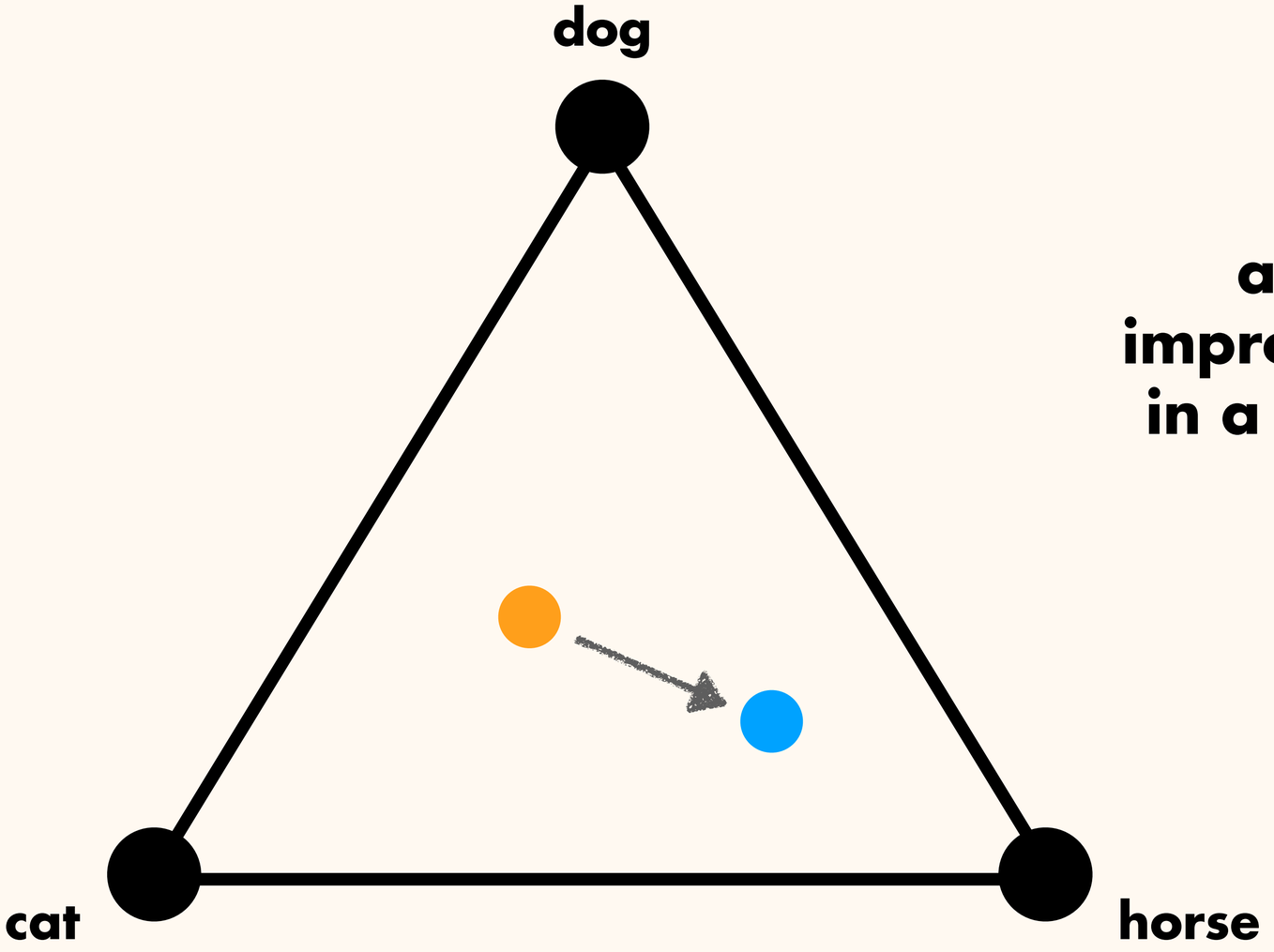


Combining these two ways works even better.



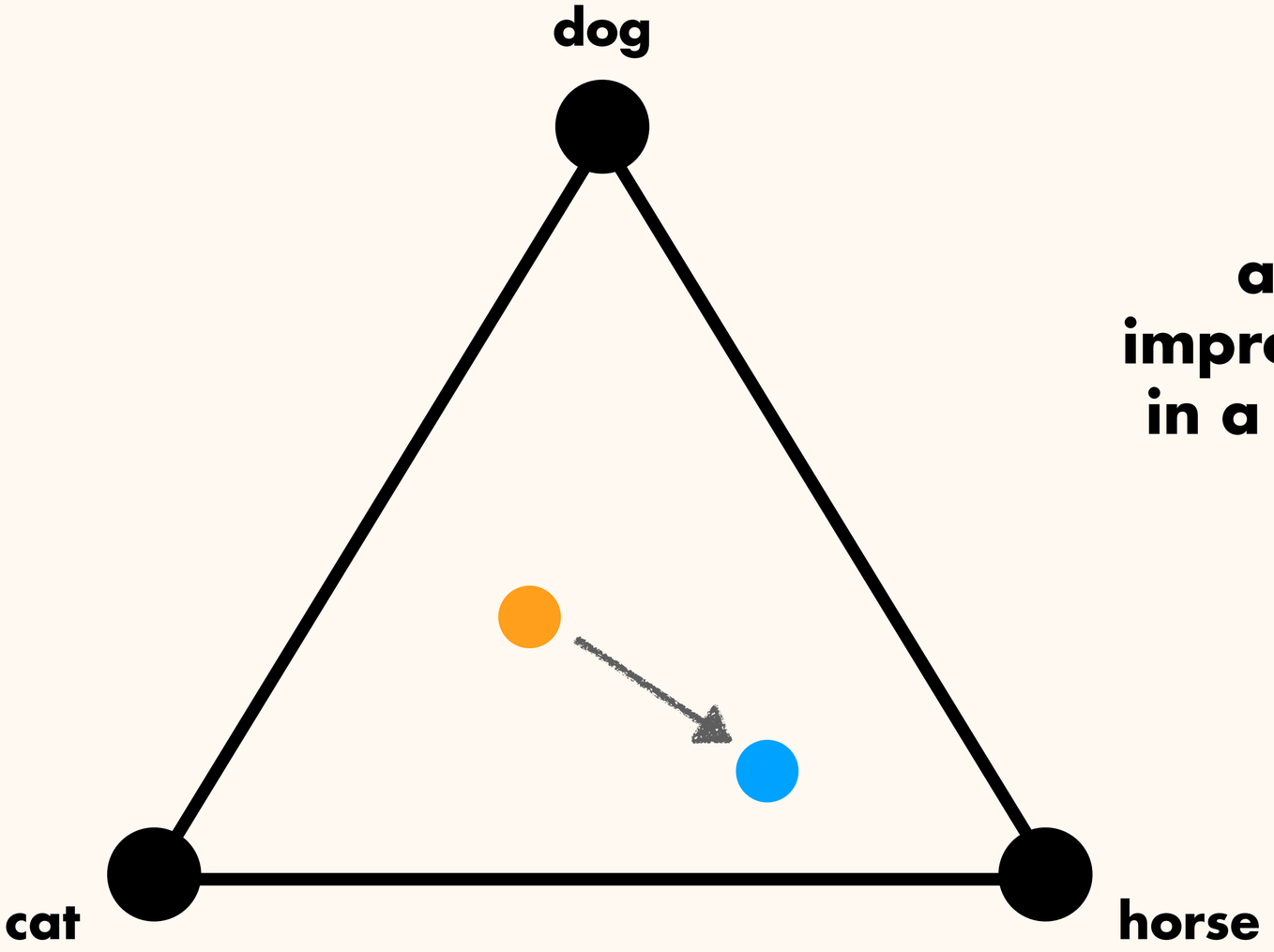
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



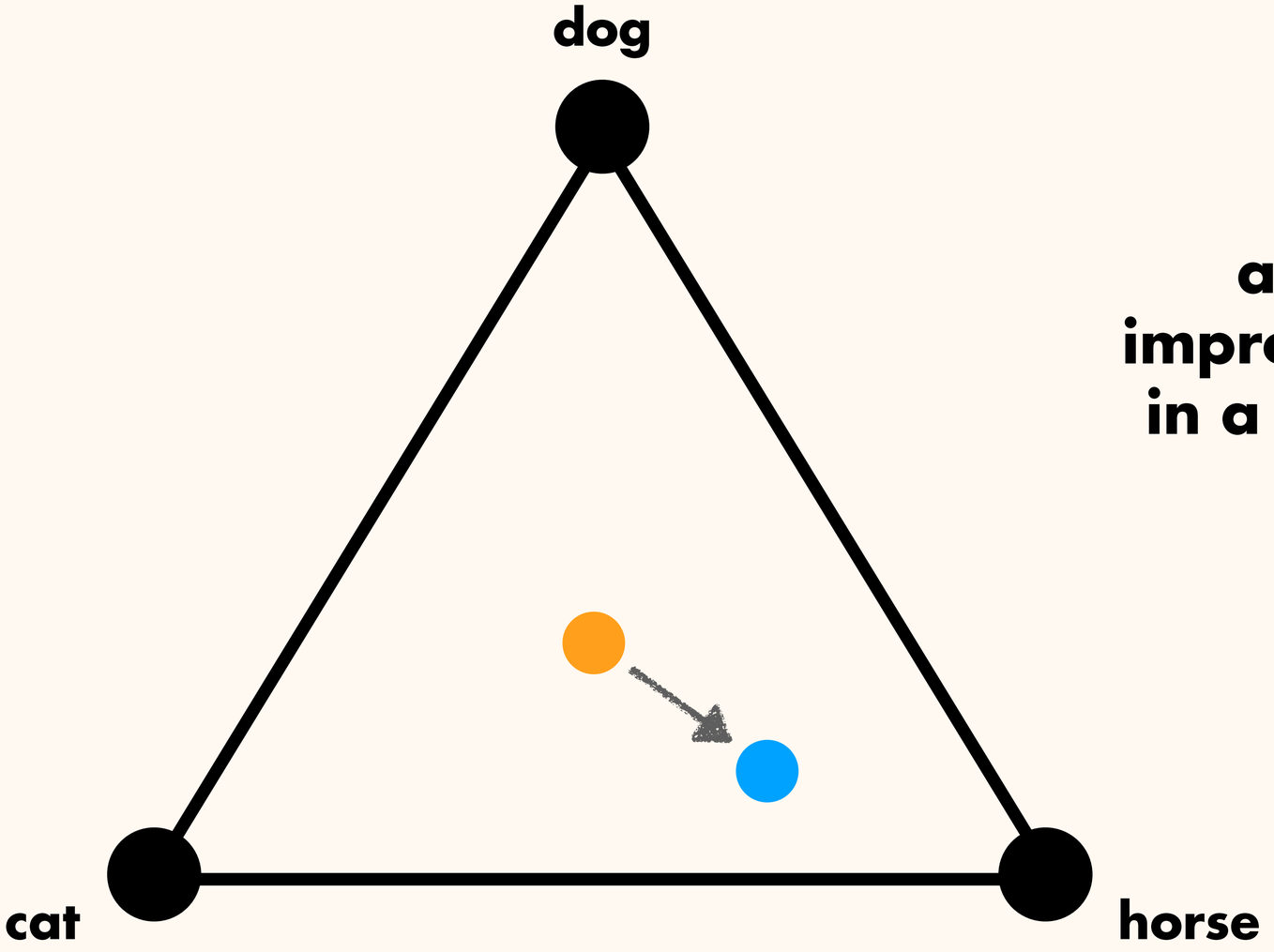
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



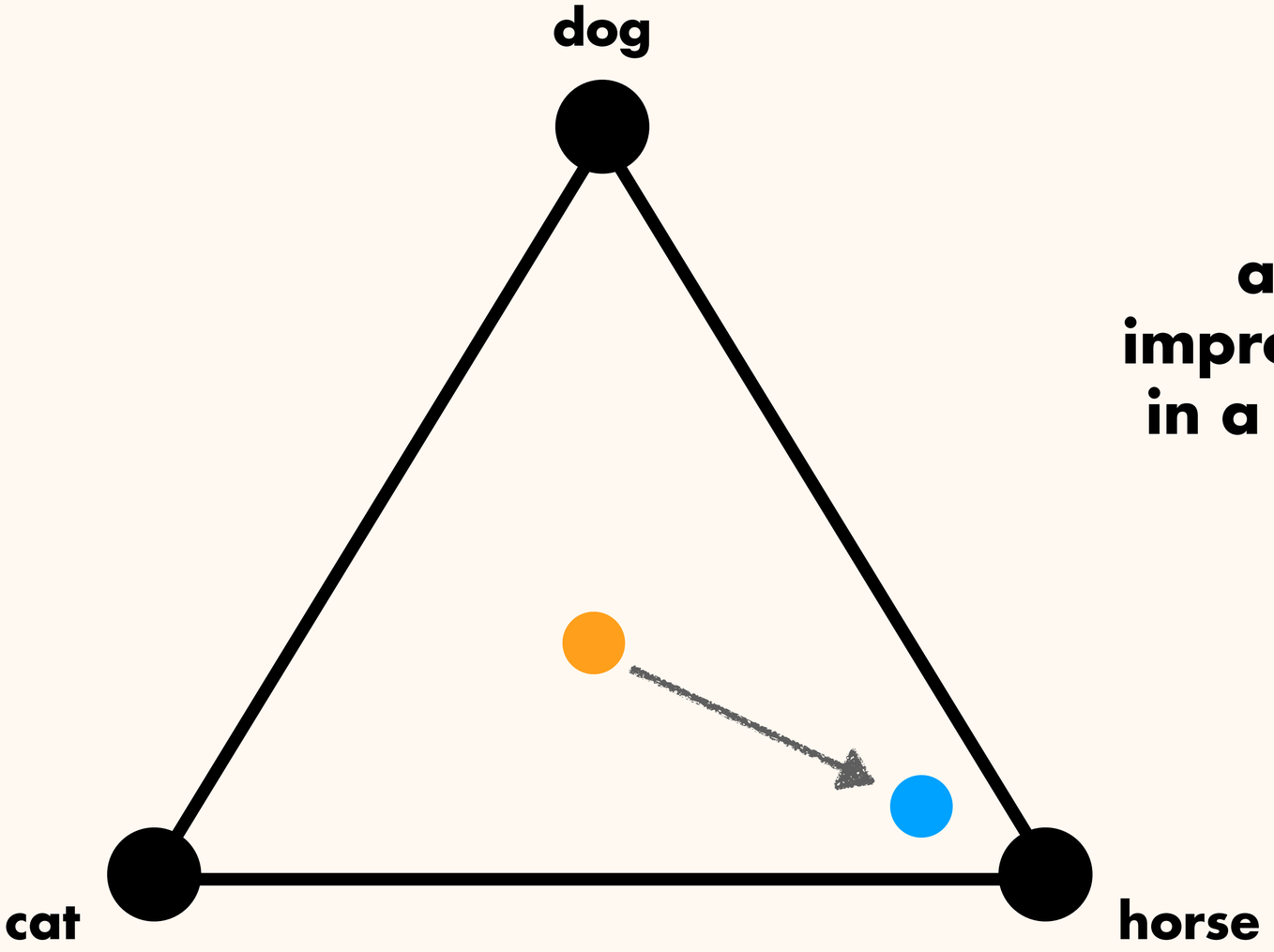
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



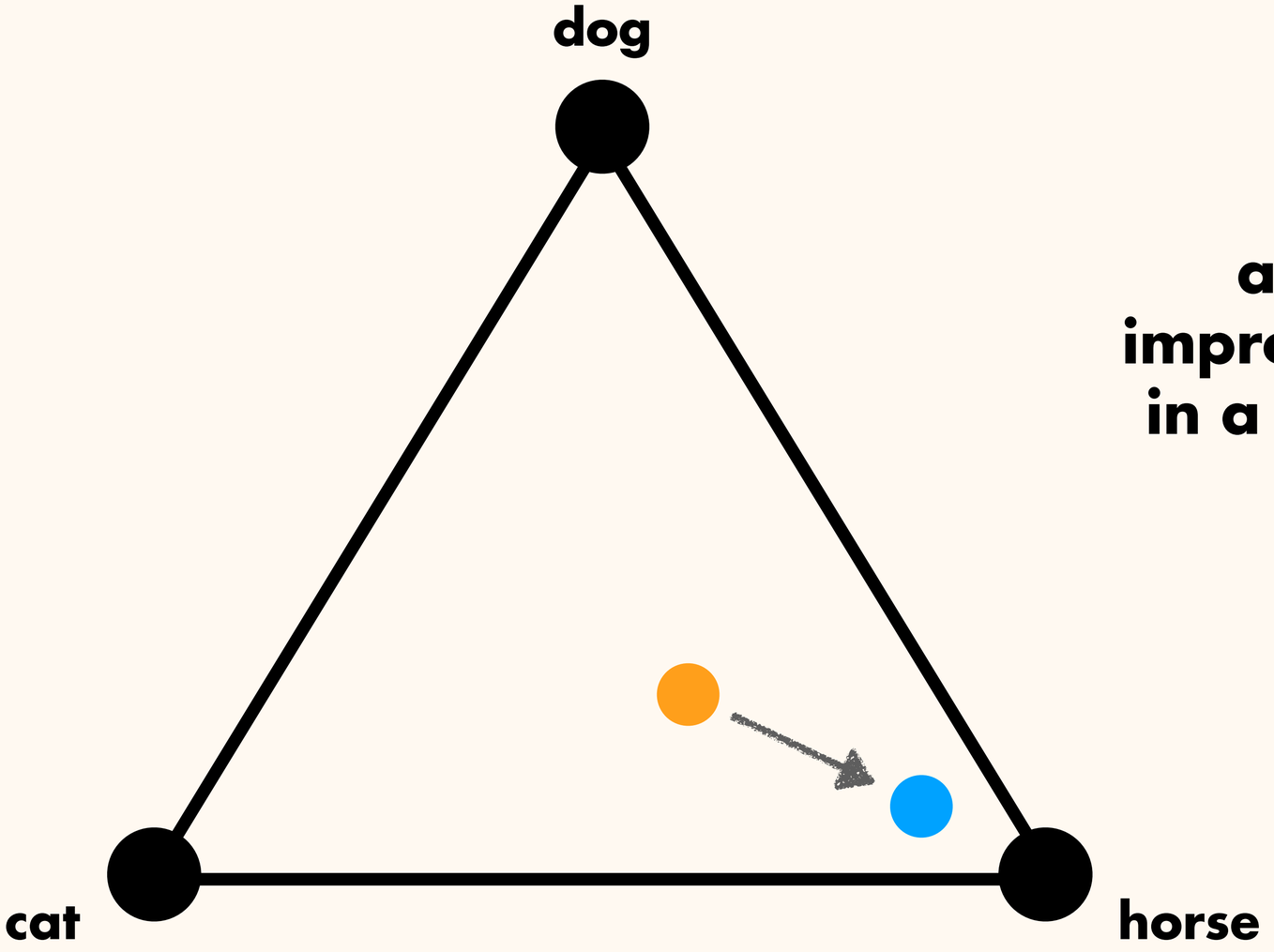
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



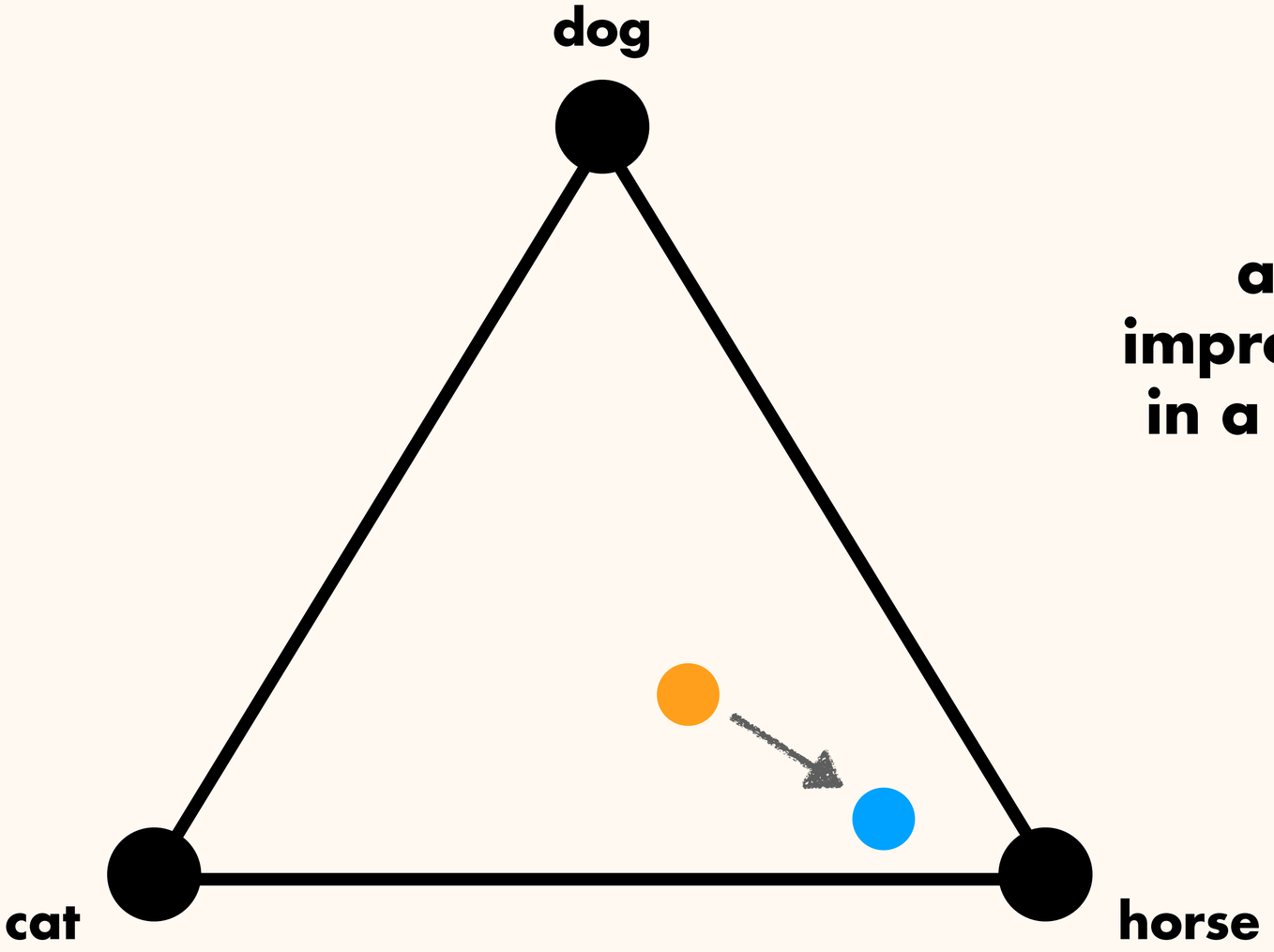
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



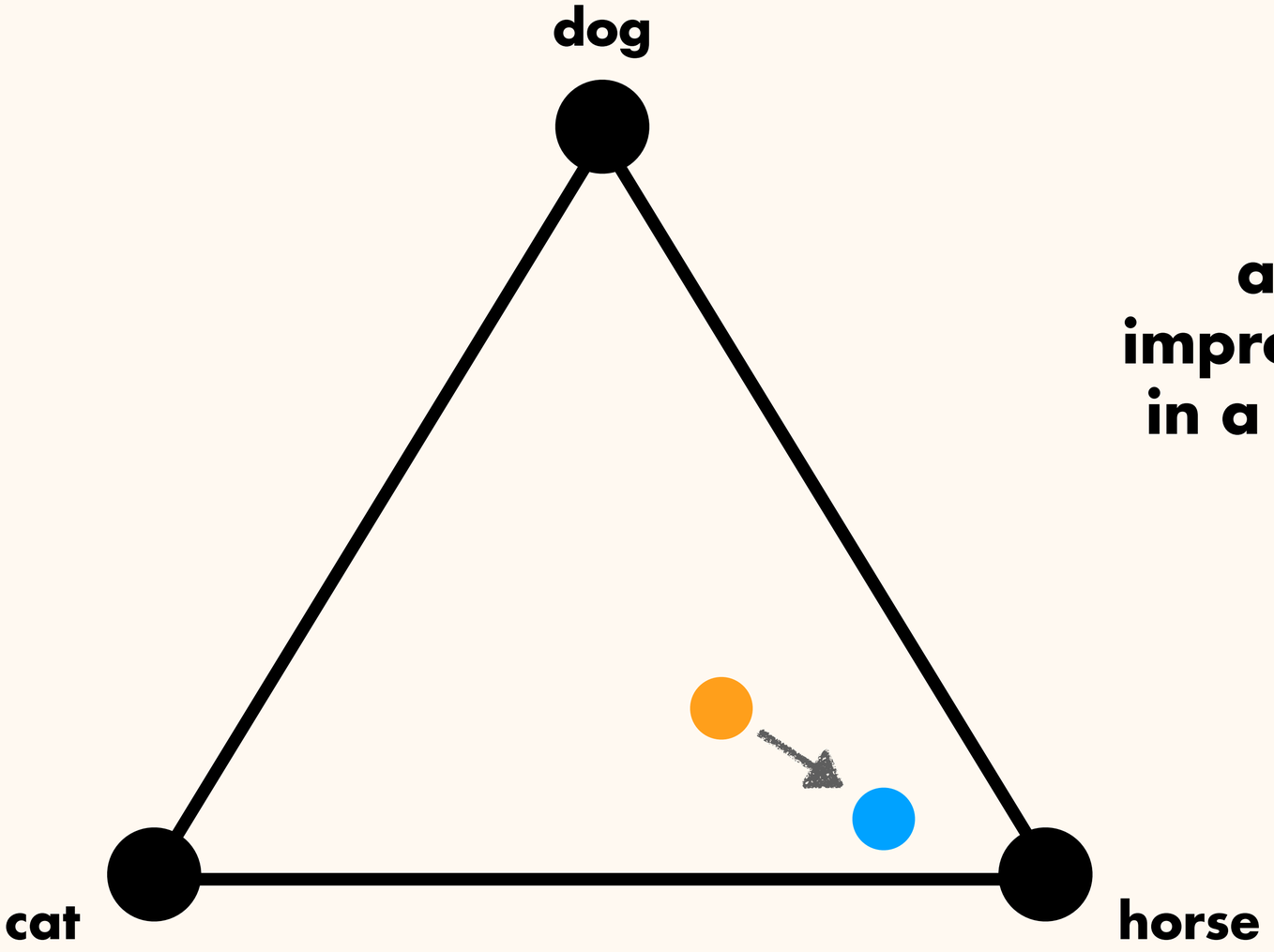
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



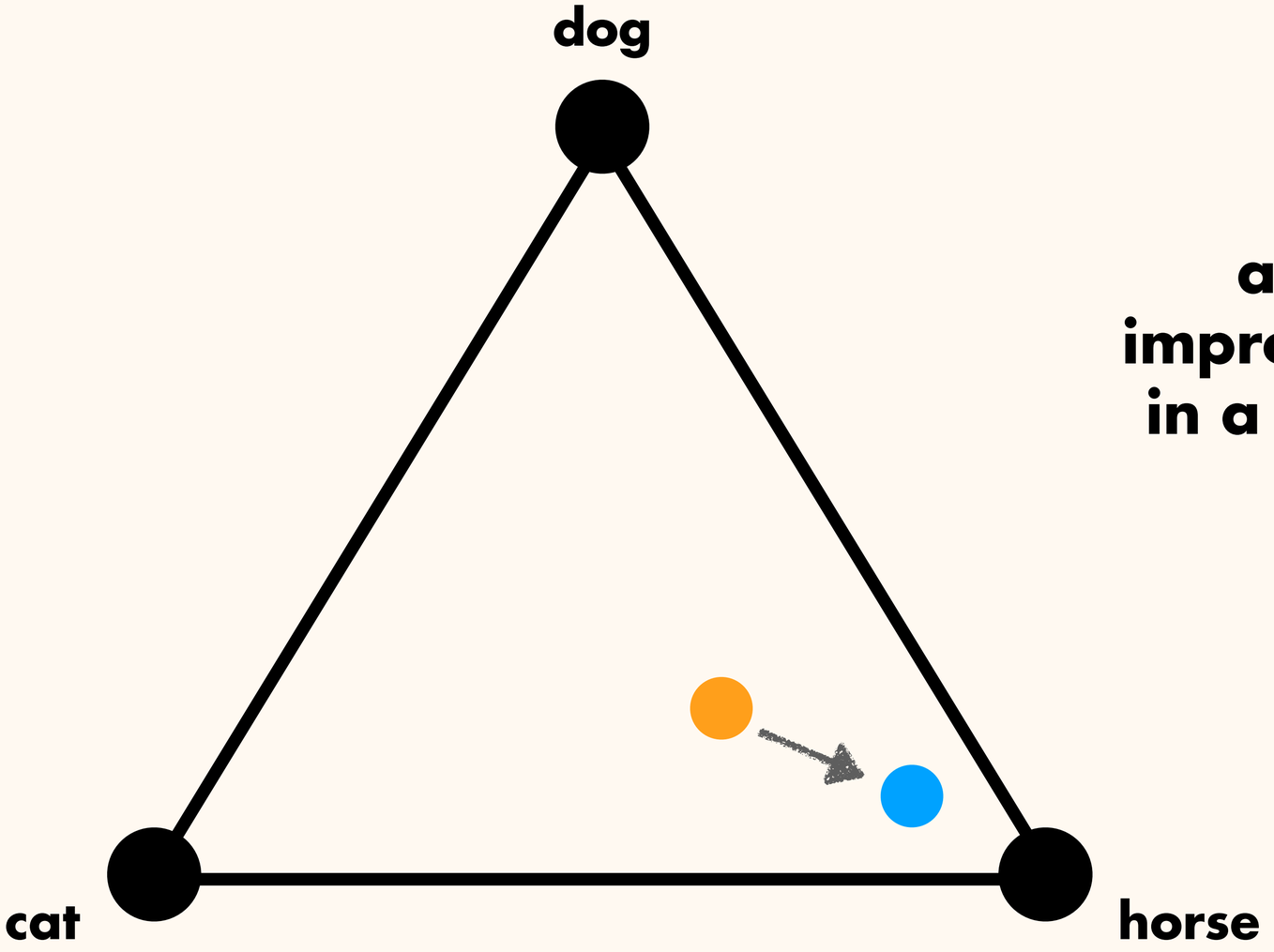
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



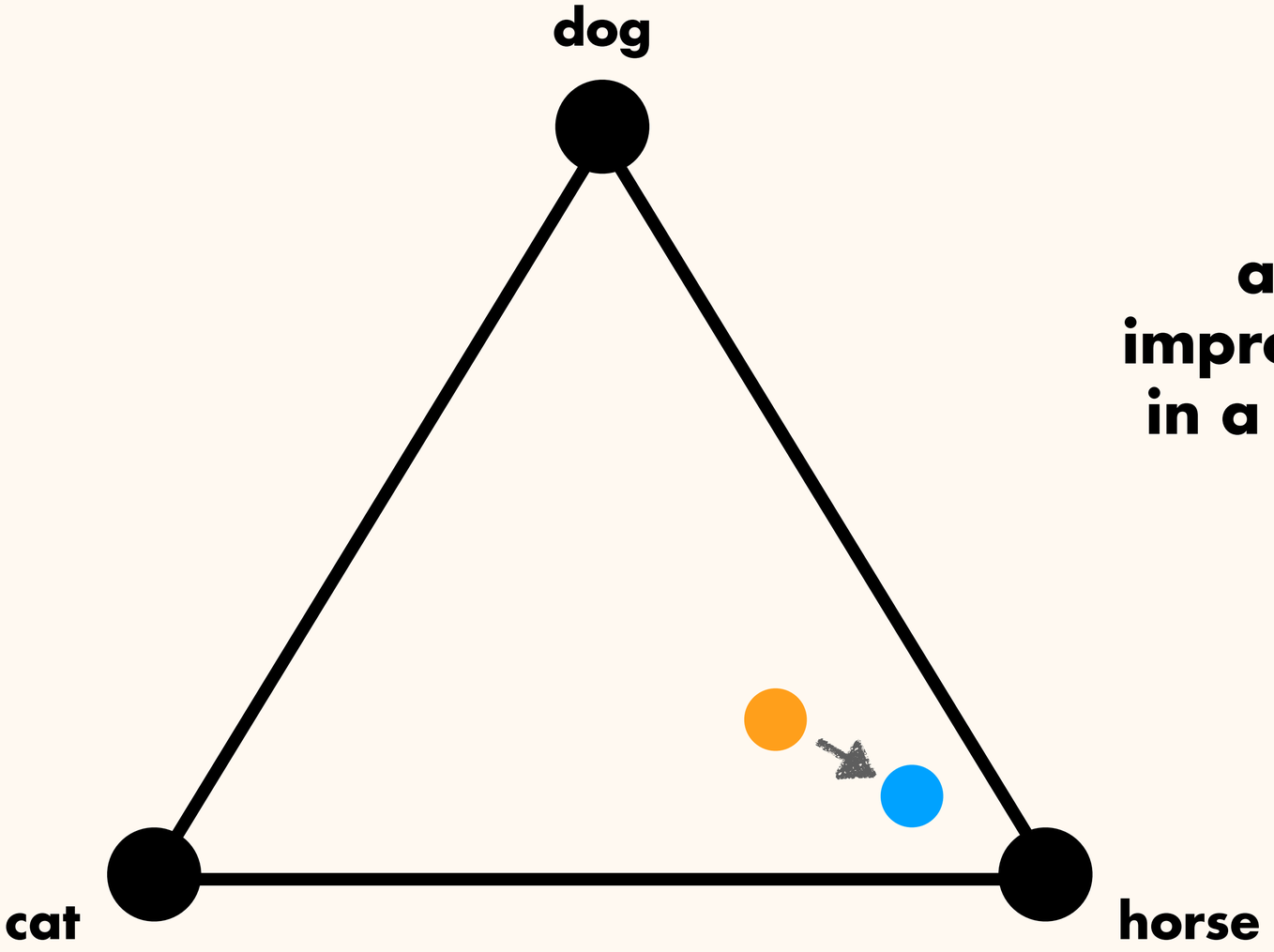
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY



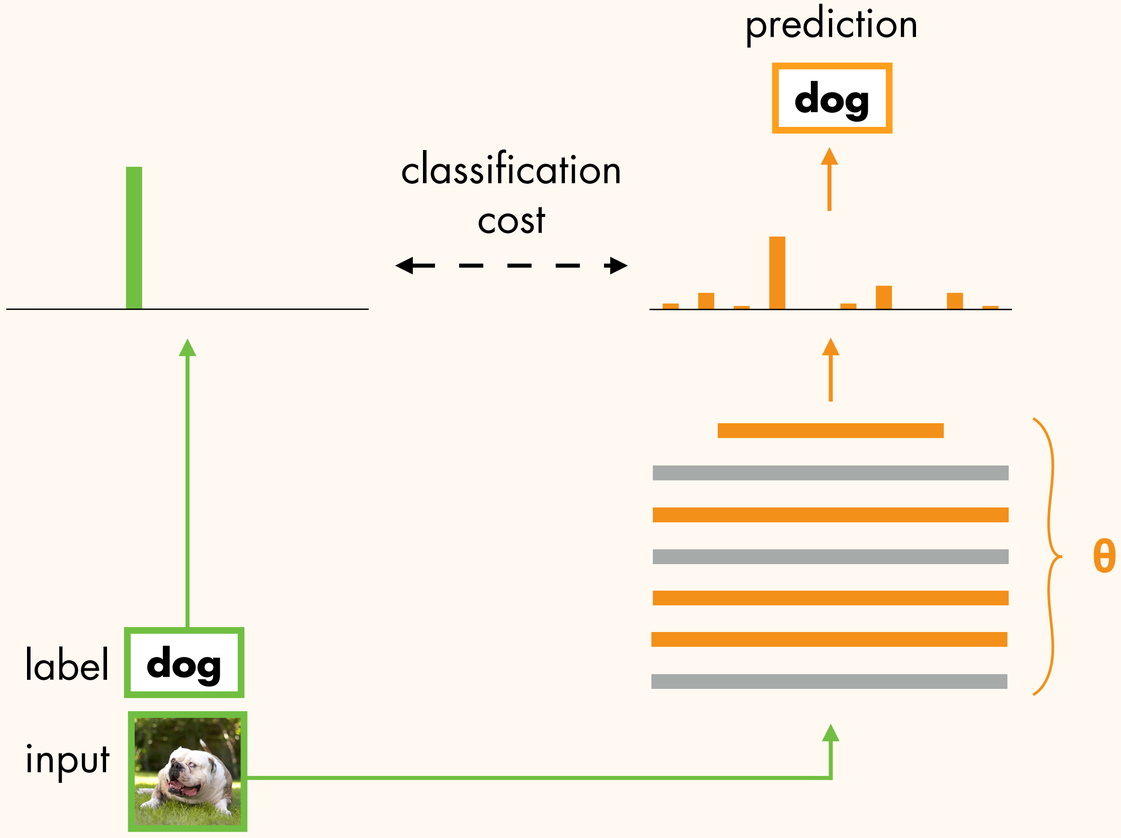
The student and the teacher improve each other in a virtuous cycle.

THE CURIOUS AI COMPANY

HOW TO START USING
THE MEAN TEACHER

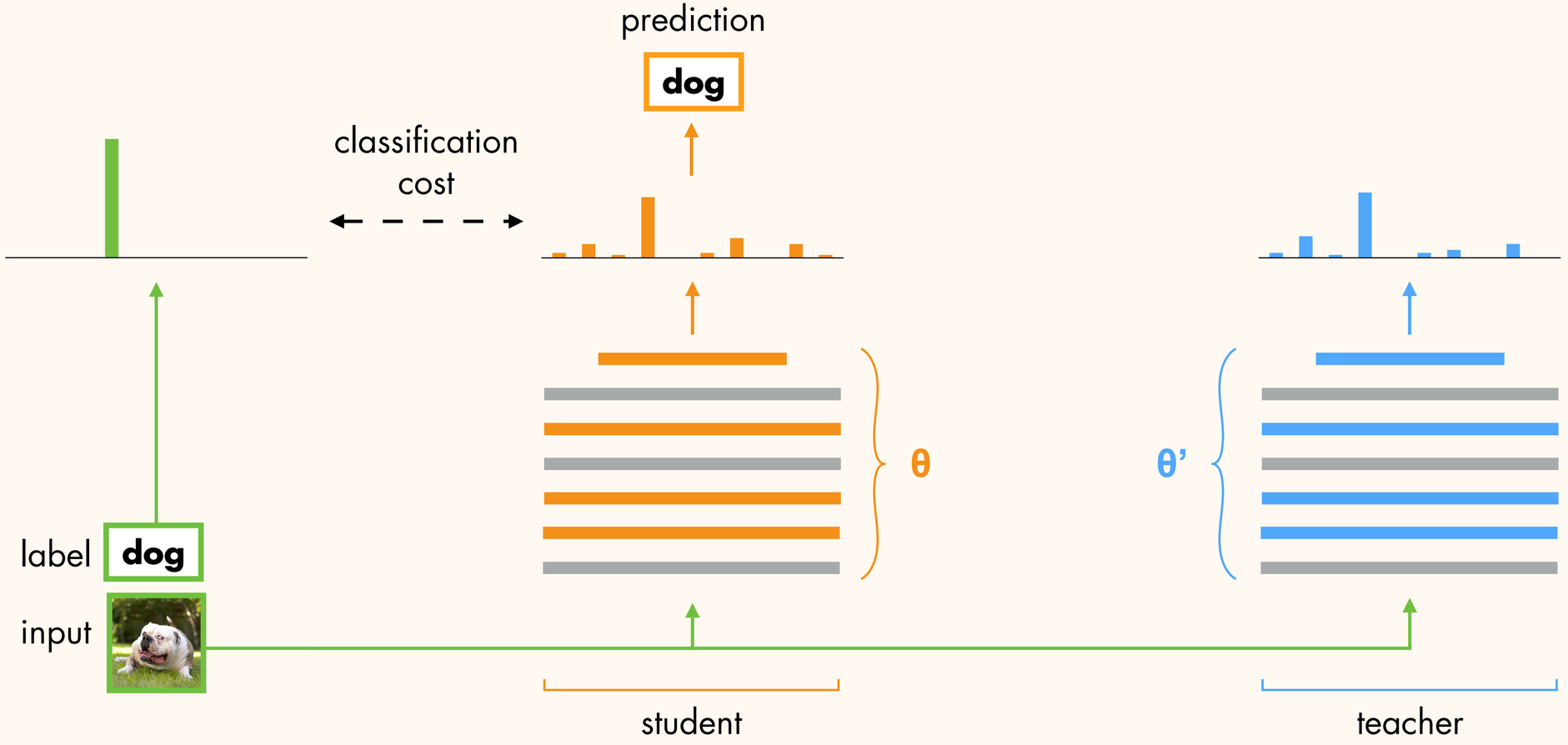
THE CURIOUS AI COMPANY

Take a supervised model.



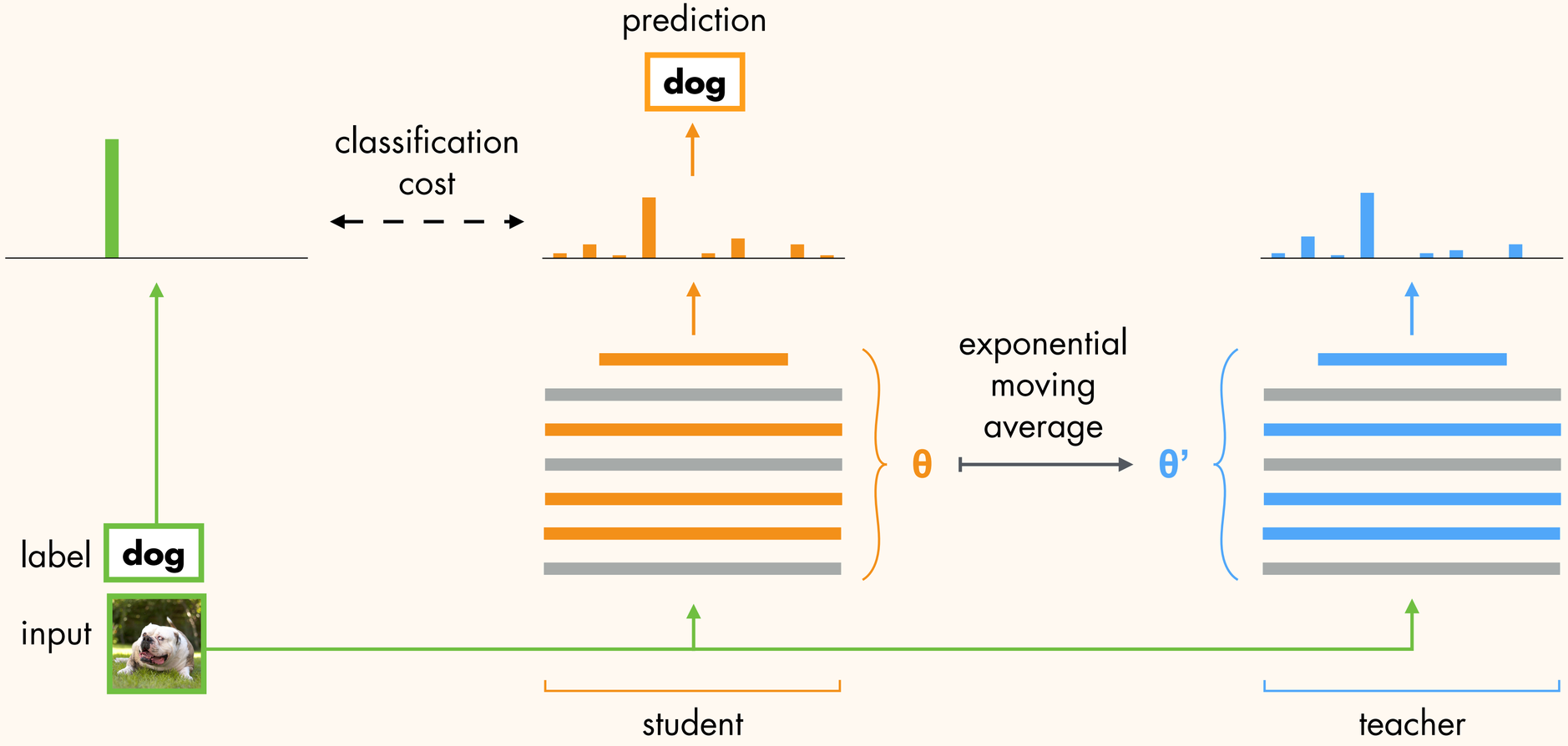
THE CURIOUS AI COMPANY

Make a copy of it.



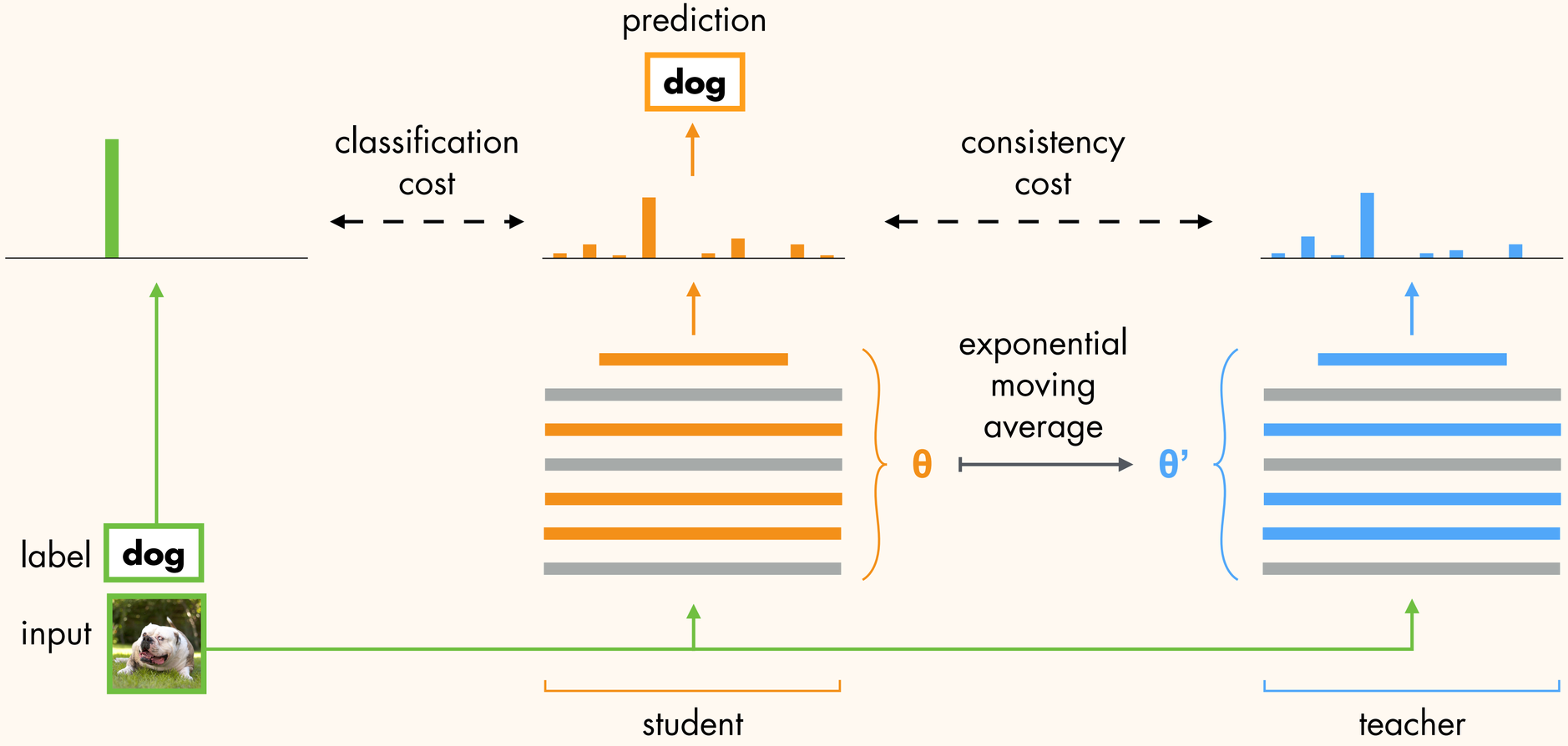
THE CURIOUS AI COMPANY

Update teacher weights after each training step.



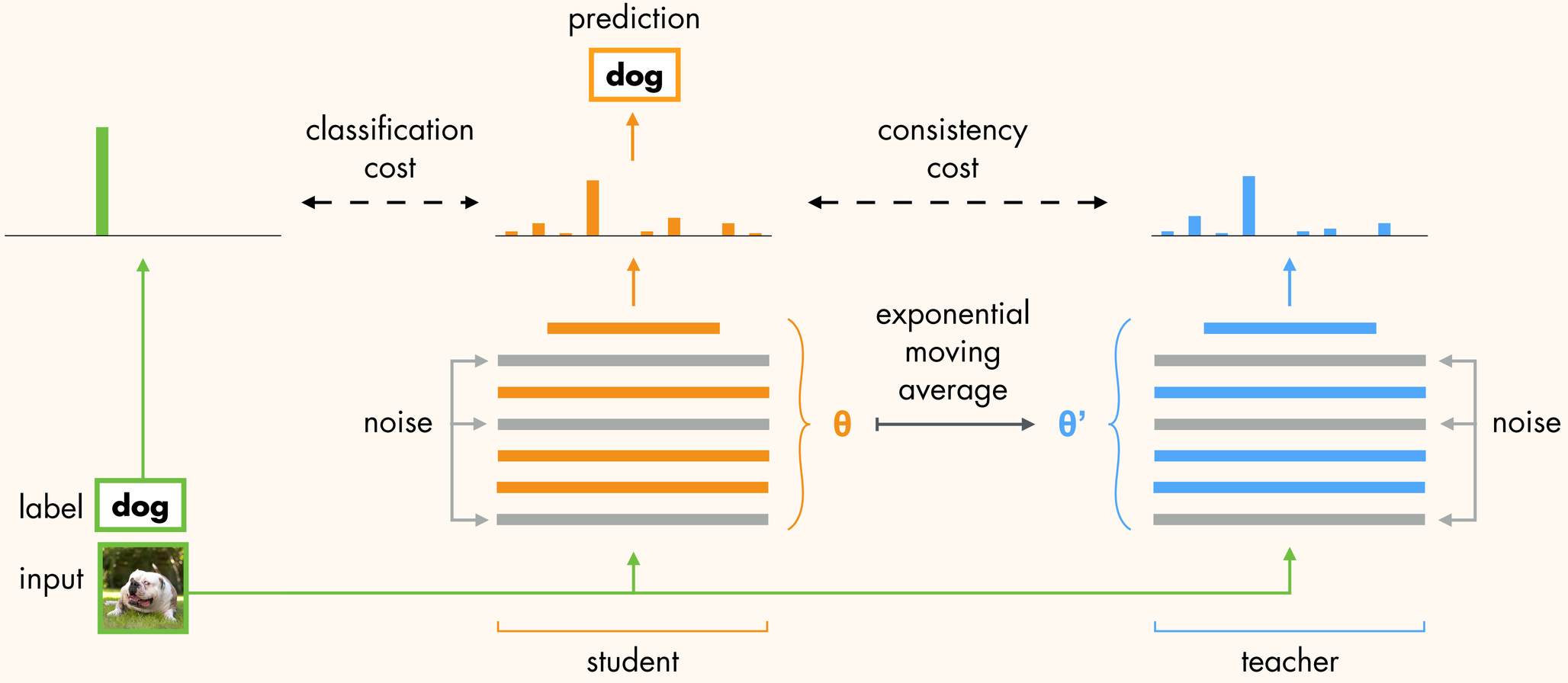
THE CURIOUS AI COMPANY

Add a cost between the two predictions.

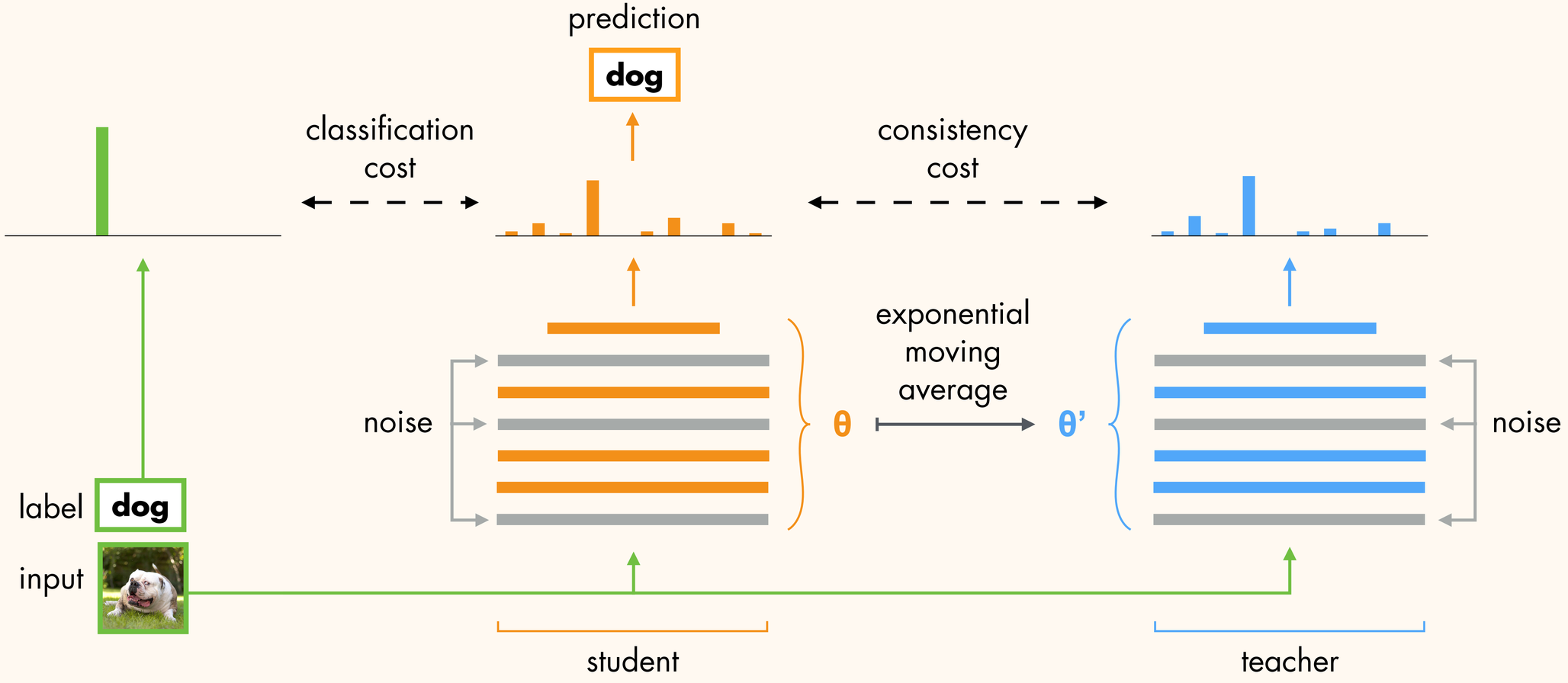


THE CURIOUS AI COMPANY

Maybe add some noise.



Start using it for semi-supervised learning.



THE CURIOUS AI COMPANY

Optimization criteria-Mean Teacher

- Classification cost - Cross-entropy loss between student softmax output and one-hot labels
- Consistency cost, $J(\theta)$ - MSE between the student and teacher softmax outputs
  - Approximate $J(\theta)$ by sampling η and η' at each training step with SGD

$$J(\theta) = E_{x,\eta',\eta} \| f(x, \theta', \eta') - f(x, \theta, \eta) \|^2$$

$$\theta_{t'} = \alpha \theta_{t-1} + (1 - \alpha) \theta_t$$

- Total cost : Weighted average of classification cost and consistency cost
  - Weight for classification : Expected number of labeled examples per mini-batch

RESULTS

THE CURIOUS COMPANY AI

Experimental setup

- 13-layer ConvNet architecture
  - Applied mean-only batch-normalization and weight normalization on convolutional and softmax layers
  - Used leaky-ReLU with α = 0.1
- 12 block(26-layer) ResNet with Shake-shake regularization
- 3 types of Noise
  - random translations and horizontal flips
  - Gaussian noise on the input layer
  - Dropout applied withing the network
- Datasets
  - Street View House Numbers(SVHN) : 73257 training, 26032 test
  - CIFAR-10 : 50000 training, 10000 test

Table 6: The convolutional network architecture we used in the experiments.

Layer	Hyperparameters
Input	32×32 RGB image
Translation	Randomly $\{\Delta x, \Delta y\} \sim [-2, 2]$
Horizontal flip[a]	Randomly $p = 0.5$
Gaussian noise	$\sigma = 0.15$
Convolutional	128 filters, 3×3, *same* padding
Convolutional	128 filters, 3×3, *same* padding
Convolutional	128 filters, 3×3, *same* padding
Pooling	Maxpool 2×2
Dropout	$p = 0.5$
Convolutional	256 filters, 3×3, *same* padding
Convolutional	256 filters, 3×3, *same* padding
Convolutional	256 filters, 3×3, *same* padding
Pooling	Maxpool 2×2
Dropout	$p = 0.5$
Convolutional	512 filters, 3×3, *valid* padding
Convolutional	256 filters, 1×1, *same* padding
Convolutional	128 filters, 1×1, *same* padding
Pooling	Average pool ($6 \times 6 \rightarrow 1 \times 1$ pixels)
Softmax	Fully connected $128 \rightarrow 10$

[a] Not applied on SVHN experiments

Baselines

Apply unsupervised loss component to both labelled and unlabeled examples

- Ladder networks - Pi model (Laine & Aila, 2016) : Ensembles predicitions of the model under 2 different perturbations of input data and different dropout conditions
- Temporal Ensembling - Ensemble predictions of a model at different timesteps

Baelines	$J(\theta) = E_{x,\eta',\eta} \| f(x, \theta', \eta') - f(x, \theta, \eta) \|^2$
Π model	$\theta' = \theta$
Temporal Ensembling	weighted average of successive predictions (θ' is not optimized)
Mean Teacher (Our)	Expected moving average of of succesive model weights (θ' is not optimized)

Laine, Samuli and Aila, Timo. Temporal Ensembling for Semi-Supervised Learning. arXiv:1610.02242 [cs], October 2016. arXiv: 1610.02242.

Table 5: Error rate percentage on SVHN and CIFAR-10 over 10 runs, including the results without input augmentation. We use exponential moving average weights in the evaluation of all our models. All the comparison methods use a 13-layer ConvNet architecture similar to ours and augmentation similar to ours, expect GAN, which does not use augmentation.

SVHN	250 labels	500 labels	1000 labels	all labels[a]
GAN[b]		18.44 ± 4.8	8.11 ± 1.3	
Π model[c]		6.65 ± 0.53	4.82 ± 0.17	2.54 ± 0.04
Temporal Ensembling[c]		5.12 ± 0.13	4.42 ± 0.16	2.74 ± 0.06
VAT+EntMin[d]			**3.86**	
Ours				
Supervised-only[e]	27.77 ± 3.18	16.88 ± 1.30	12.32 ± 0.95	2.75 ± 0.10
Π model	9.69 ± 0.92	6.83 ± 0.66	4.95 ± 0.26	2.50 ± 0.07
Mean Teacher	$\mathbf{4.35 \pm 0.50}$	$\mathbf{4.18 \pm 0.27}$	3.95 ± 0.19	$\mathbf{2.50 \pm 0.05}$
Without augmentation				
Supervised-only[e]	36.26 ± 3.83	19.68 ± 1.03	14.15 ± 0.87	3.04 ± 0.04
Π model	10.36 ± 0.94	7.01 ± 0.29	5.73 ± 0.16	2.75 ± 0.08
Mean Teacher	5.85 ± 0.62	5.45 ± 0.14	5.21 ± 0.21	2.77 ± 0.09

CIFAR-10	1000 labels	2000 labels	4000 labels	all labels[a]
GAN[b]			18.63 ± 2.32	
Π model[c]			12.36 ± 0.31	$\mathbf{5.56 \pm 0.10}$
Temporal Ensembling[c]			12.16 ± 0.31	5.60 ± 0.10
VAT+EntMin[d]			10.55	
Ours				
Supervised-only[e]	46.43 ± 1.21	33.94 ± 0.73	20.66 ± 0.57	5.82 ± 0.15
Π model	27.36 ± 1.20	18.02 ± 0.60	13.20 ± 0.27	6.06 ± 0.11
Mean Teacher	21.55 ± 1.48	$\mathbf{15.73 \pm 0.31}$	12.31 ± 0.28	5.94 ± 0.15
Mean Teacher ResNet	$\mathbf{10.08 \pm 0.41}$		$\mathbf{6.28 \pm 0.15}$	
Without augmentation				
Supervised-only[e]	48.38 ± 1.07	36.07 ± 0.90	24.47 ± 0.50	7.43 ± 0.06
Π model	32.18 ± 1.33	23.92 ± 1.07	17.08 ± 0.32	7.00 ± 0.20
Mean Teacher	30.62 ± 1.13	23.14 ± 0.46	17.74 ± 0.30	7.21 ± 0.24

[a] 4 runs [b] Salimans et al. [25] [c] Laine & Aila [13] [d] Miyato et al. [16]
[e] Only labeled examples and only classification cost

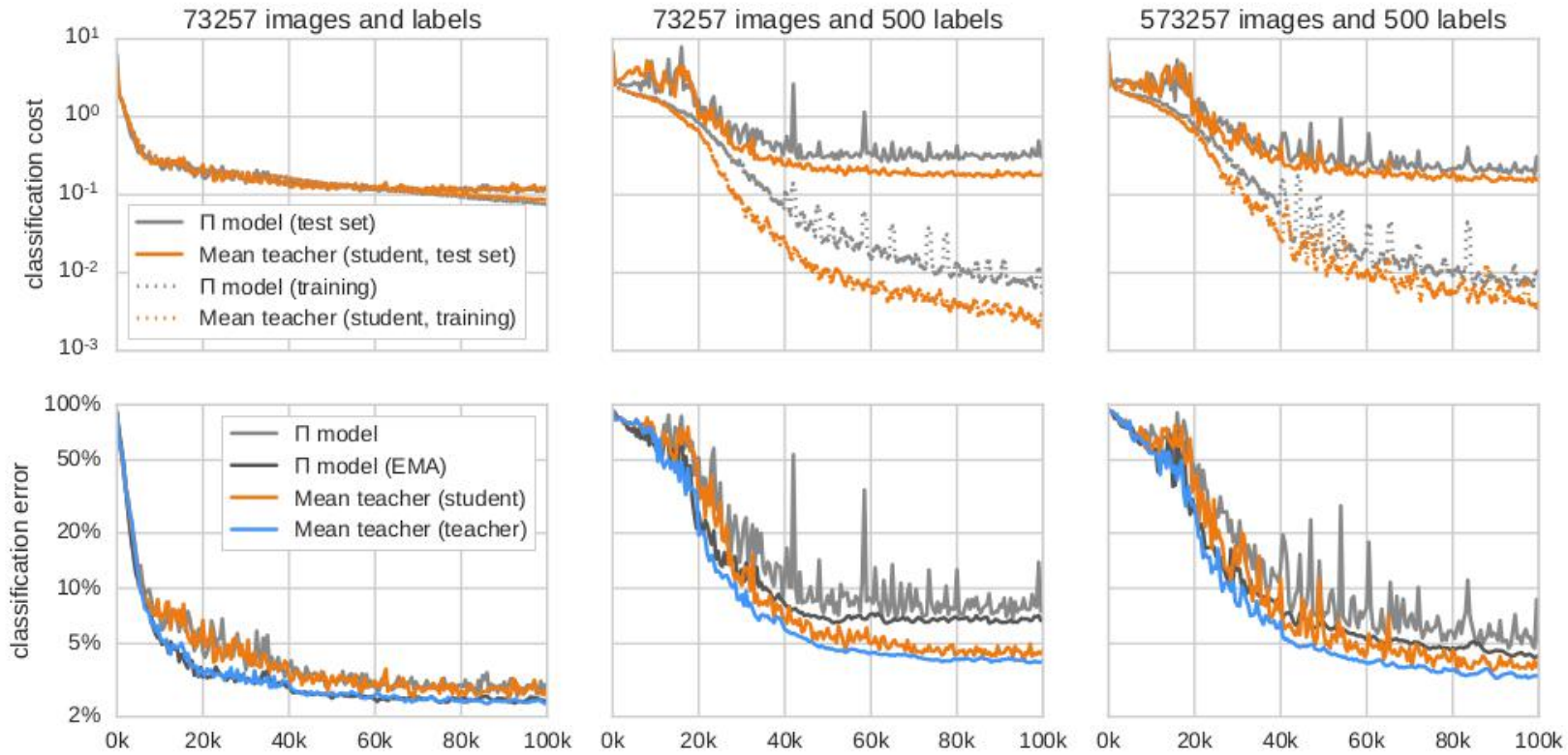
1. MT performs better when the number of labelled examples is less

2. VAT+EntMin (Miyato et. al., 2017) outperforms MT in certain cases

Miyato, Takeru, Maeda, Shin-ichi, Koyama, Masanori, and Ishii, Shin. Virtual Adversarial Training: a Regularization Method for Supervised and Semi-supervised Learning. arXiv:1704.03976 [cs, stat], April 2017. arXiv: 1704.03976.

Table 3: Error percentage over 10 runs on SVHN with extra unlabeled training data.

	500 labels 73257 images	500 labels 173257 images	500 labels 573257 images
Π model (ours)	6.83 ± 0.66	4.49 ± 0.27	3.26 ± 0.14
Mean Teacher	$\mathbf{4.18 \pm 0.27}$	$\mathbf{3.02 \pm 0.16}$	$\mathbf{2.46 \pm 0.06}$



(right column) : Pi-model keeps improving for longer. MT learns faster and eventually converges to abetter result.

Fig : Smoothened classification cost (top) and classification error (bottom) of Mean Teacher and our baseline Π model on SVHN over the first 100000 training steps. In the upper row, the training classification costs are measured using only labeled data.

Other experiments

- Instead of using MSE(Laine & Aila, 2016) for calculating consistency cost, we use KL-divergence

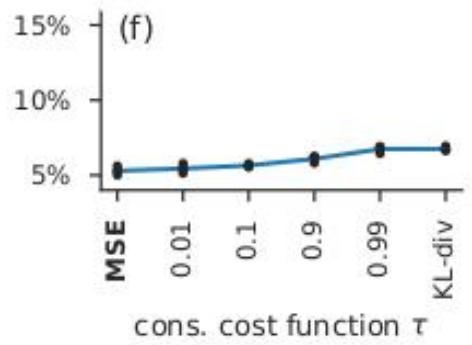


Figure 5: Copy of Figure 4 (f) in the main text. Validation error on 250-label SVHN over four runs and their mean, when varying the consistency cost shape hyperparameter τ between mean squared error ($\tau = 0$) and KL-divergence ($\tau = 1$).

MSE performs better than KL-divergence or C_T for different values of T

Laine, Samuli and Aila, Timo. Temporal Ensembling for Semi-Supervised Learning. arXiv:1610.02242 [cs], October 2016. arXiv: 1610.02242.

Related works

- Regularize intermediate representations effectively in DL
  - adverserial training, dropout, DropConnect, Stochastic Depth, Swapout
- Training the model predictions to be consistent to perturbation
  - Denoising Source Separation framework, Ladder Networks(T-model, Pi-model), Virtual Adverserial training
- Teacher model training a student model
  - Model compression and distillation
- Consistency regularization : Like label propagation, but somewhat different !
  - LP requires predefined distance metric
  - Employ a learned distance metric and aid the network to learn a better distance metric

Conclusion

- Mean Teacher(MT) helps when the labels are scarce
  - When using only 500 labels, MT learns faster and continues training for much longer than pi-model
  - In all-labeled case, Mean teacher and pi model perform identically
- MT uses unlabeled training data more efficiently than Pi model
  - With 500k examples extra unlabeled examples, MT learns faster and end eventually converges to a better result
- Works with large datasets and on-line learning
  - Uses moving average of model weights or parameters instead of label predictions

WANT TO KNOW MORE?

PAPER

Mean Teachers Are Better Role Models

Weight-averaged consistency targets improve semi-supervised deep learning results

arxiv:1703.01780

POSTER

Today 06:30 – 10:30 PM
@ Pacific Ballroom #13

SOURCE CODE

http://github.com/CuriousAI/mean-teacher

THE CURIOUS AI COMPANY