facebook

Learning Visual Features from Large Weakly Supervised Data

ECCV 2016


Laurens van der Maaten

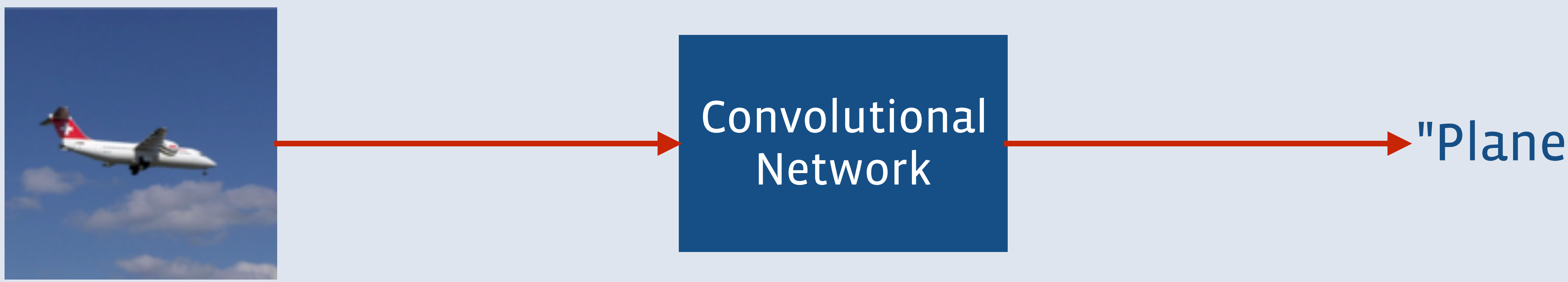

Armand Joulin


Allan Jabri


Nicolas Vasilache

Image Recognition: conventional setup

- Use large, manually curated dataset of {image,label} pairs for supervised training of large convolutional network model



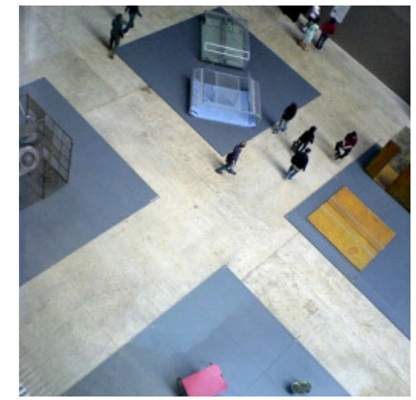
- But datasets expensive and time-consuming to build
- Hard to get beyond a few million labels

Learning from weak labels

- Facebook contains tons of data like this:

the veranda hotel portixol palma

plane approaching zrh avro regional jet rj

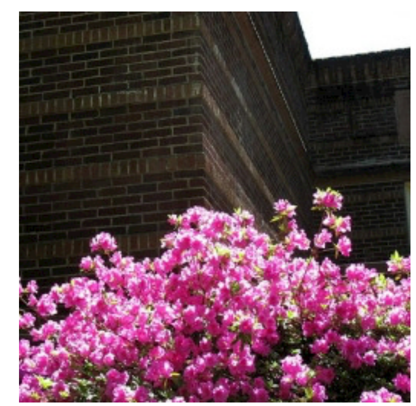


not as impressive as embankment that s for sure

student housing by lungaard tranberg architects in copenhagen click here to see where this photo was taken


Unusual is commonplace

article in the local paper about all the unusual things found at otto s home



this was another one with my old digital camera i like the way it looks for some things though slow and lower resolution than new cameras another problem is that it s a bit of a brick to carry and is a pain unless you re carrying a bag with some room it s nearly x x and weighs ounces new one is x x and weighs ounces i underexposed this one a bit did exposure bracketing script underexposure on that camera looks melty yummy gold kodak film like

Architecture

- Train convolutional network to predict words that co-occur with an image
  - Flickr 100M dataset contains ~100M photos with associated "captions"

- We treat each individual word in a photo's caption as a target for that photo
  - That is: a multi-label learning problem with extremely noise labels

- We train convolutional networks to predict the words from the images:
  - We use standard convnet architectures such as AlexNet

Loss function

- We train using multi-class logistic loss over 100K hashtags:

$$\ell(\theta, \mathbf{W}; \mathcal{D}) = \frac{-1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk} \log \left[\frac{\exp(\mathbf{w}_k^\top f(\mathbf{x}_n; \theta))}{\sum_{k'=1}^{K} \exp(\mathbf{w}_{k'}^\top f(\mathbf{x}_n; \theta))} \right]$$

  - Surprisingly, this worked better than one-versus-all losses

- Training is performed using mini-batch stochastic gradient descent:
  - We use class-uniform sampling to prevent frequent classes from dominating the visual features

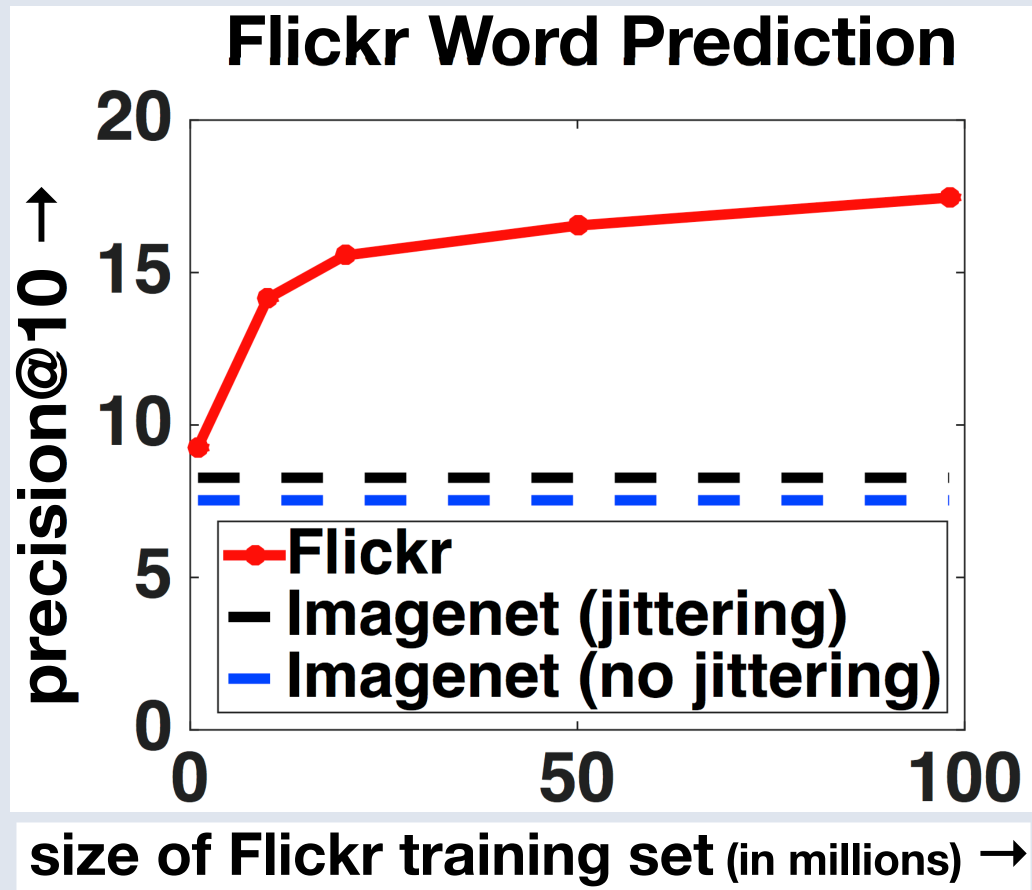
Experimental setup

- First, we train our networks on the Flickr 100M dataset
  - We perform experiments with dictionary sizes up to 100K

- We evaluate the networks in two experiments:
  - **Experiment 1:** Given a photo, predict the words
  - **Experiment 2:** Use the features learned by the convolutional networks for transfer learning to other vision tasks

Word prediction: Learning curves

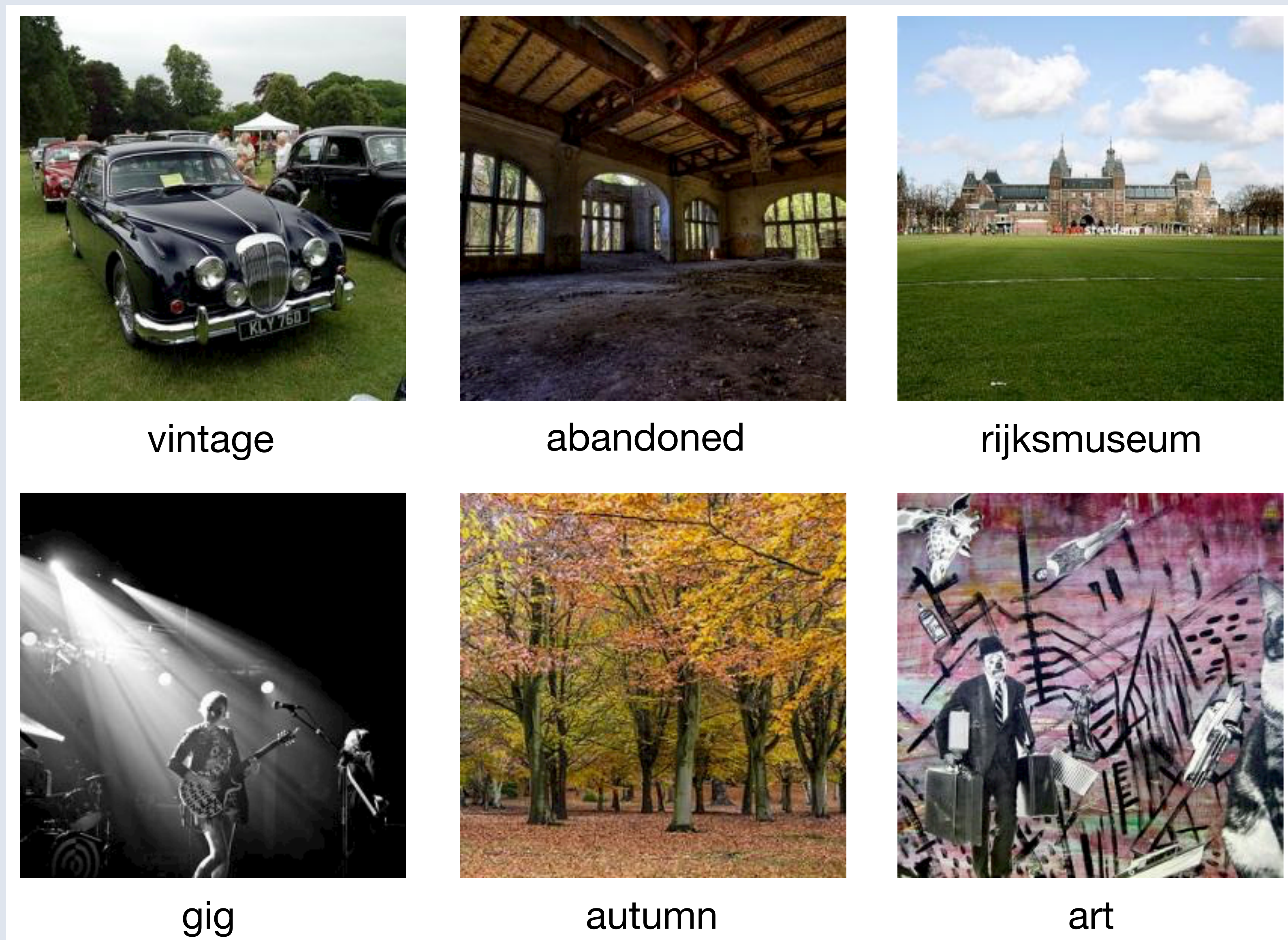
- How much data do we need to train good word prediction models?



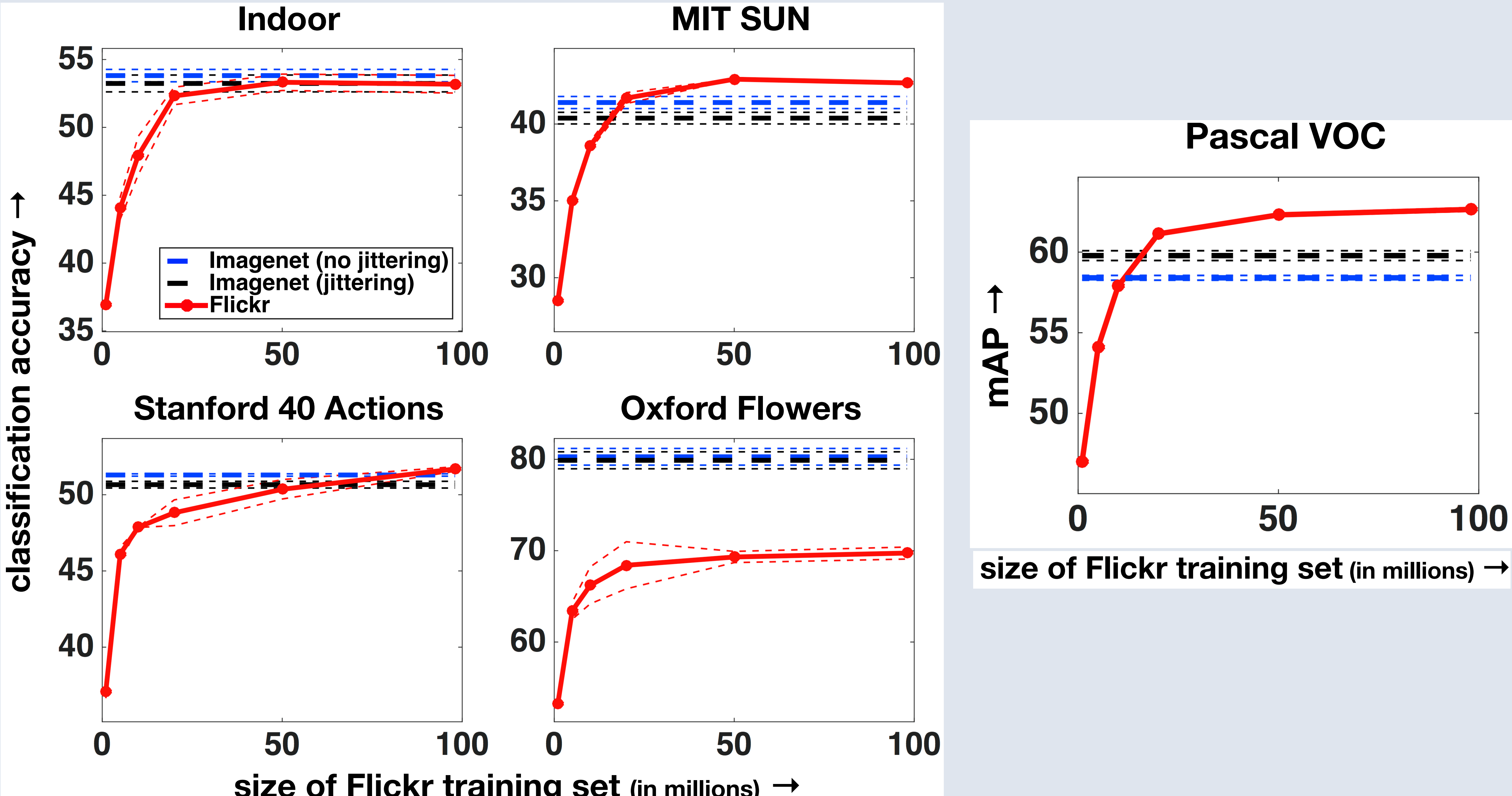
- Having tens of millions of weakly supervised images helps!

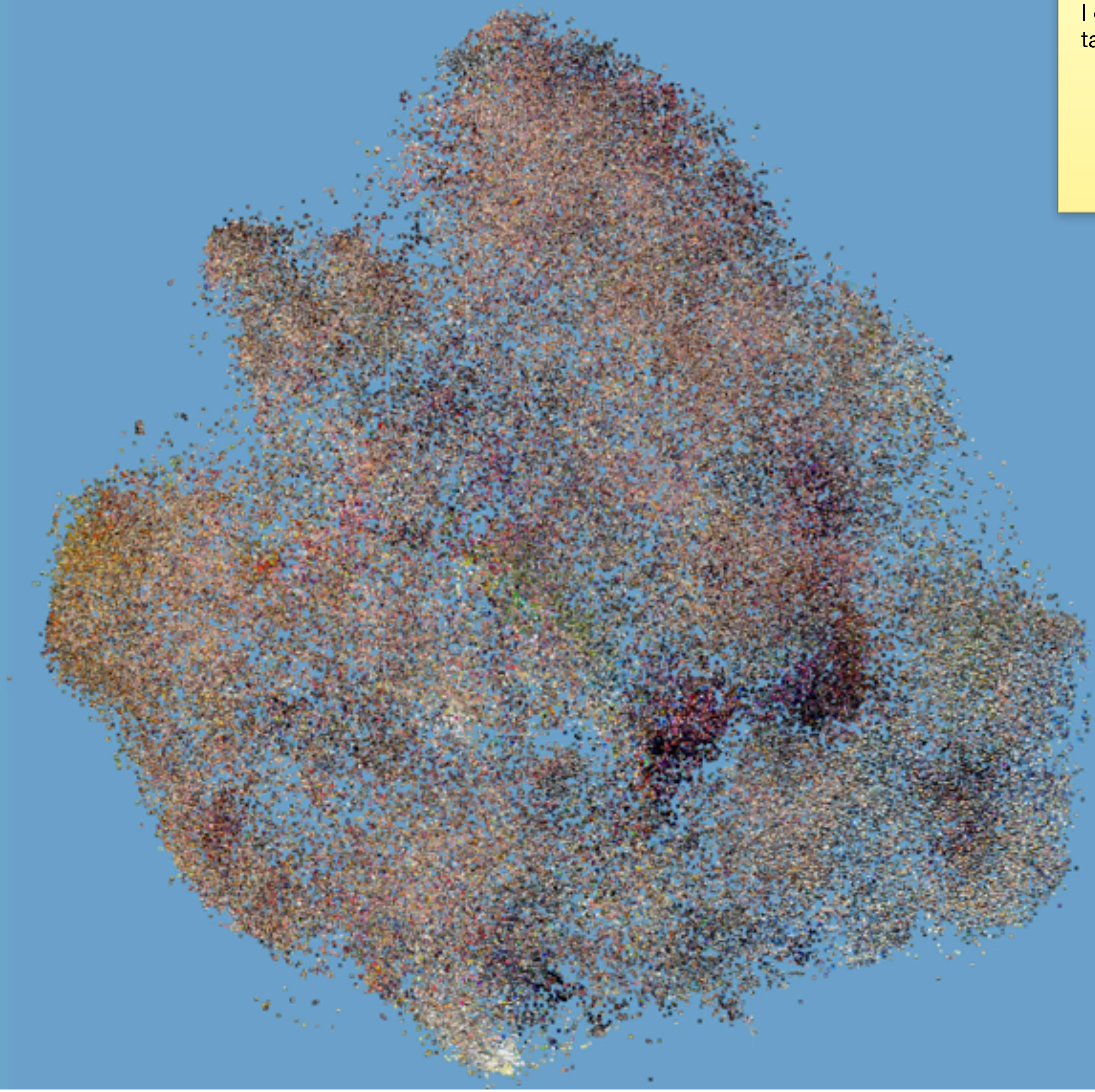
Word prediction

- Six images with high scores for arbitrary words:



Transfer Learning: Learning Curves

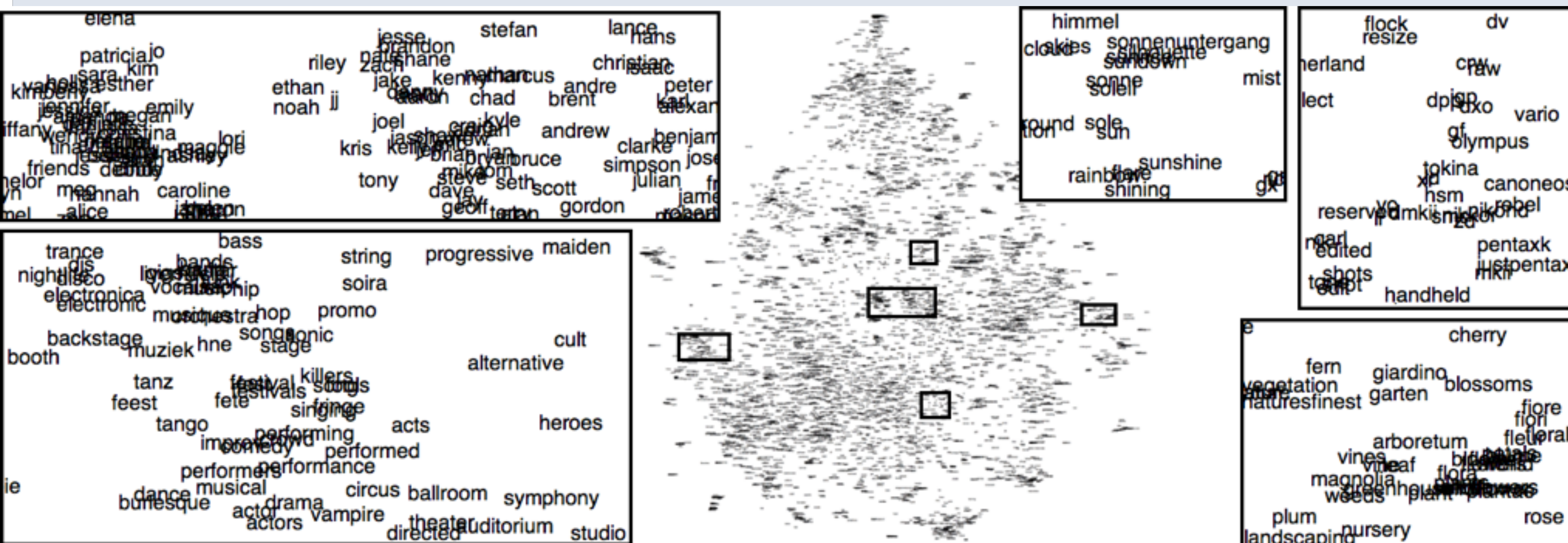




I could do a new version of this for the talk?

Analyzing the word embeddings

- Output layer of our convnets is essentially a word embedding
- This embedding has captured semantic information:



Summary

- Training with 100M images + noisy labels gives visual features comparable to 1M images + clean labels.

- Clean labels not essential for training

Random Labels????

From Ben Recht (Berkeley):