CMPT 983

Grounded Natural Language Understanding

January 19, 2022

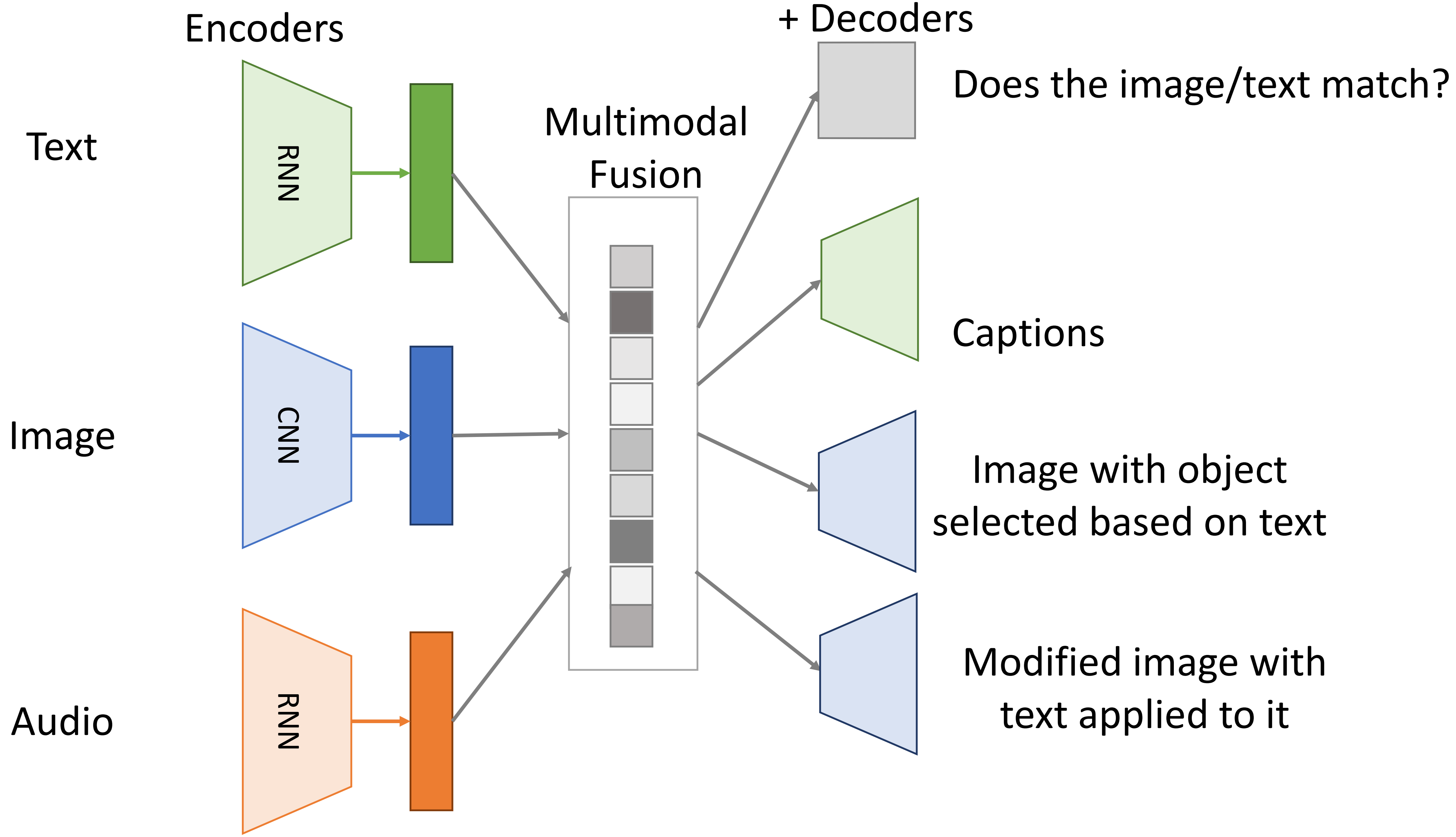
Multimodal representations

Today

- Multimodal representations
  - Joint representations
  - Correlated representations
- Applications using multimodal representations
  - Retrieval
  - Translation

Multimodal models

Multimodal models



Modeling Images

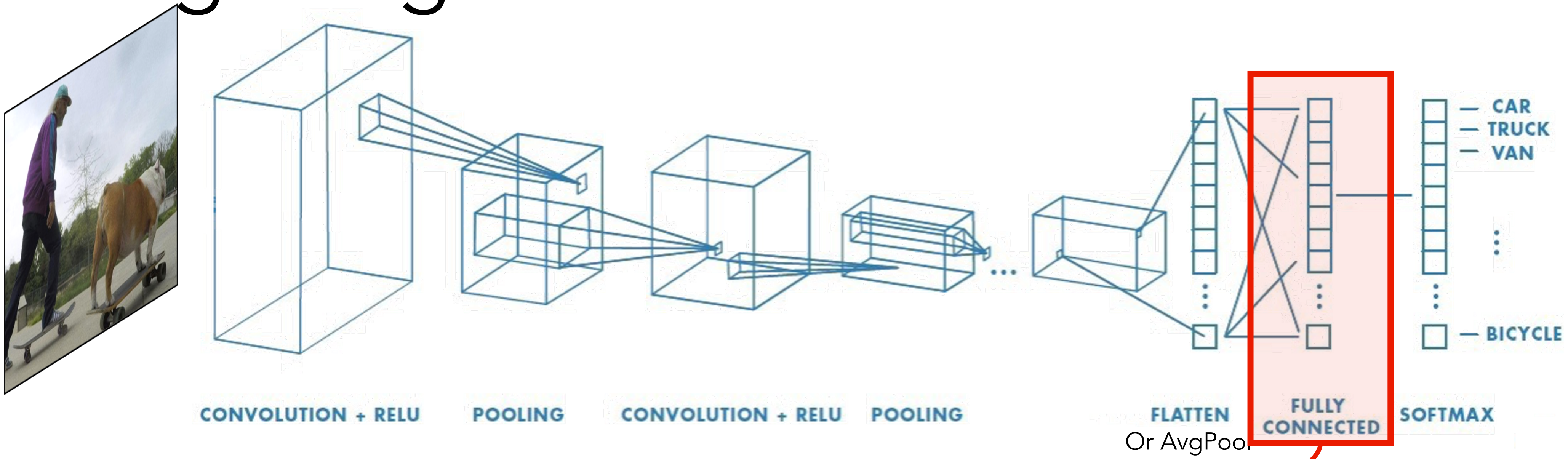
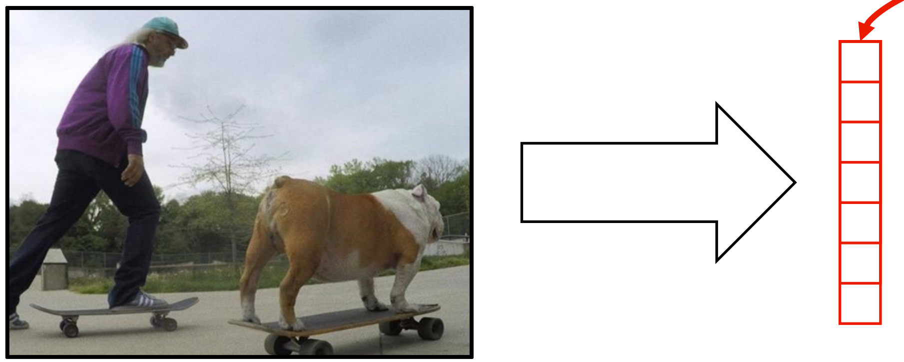


Image Level Feature

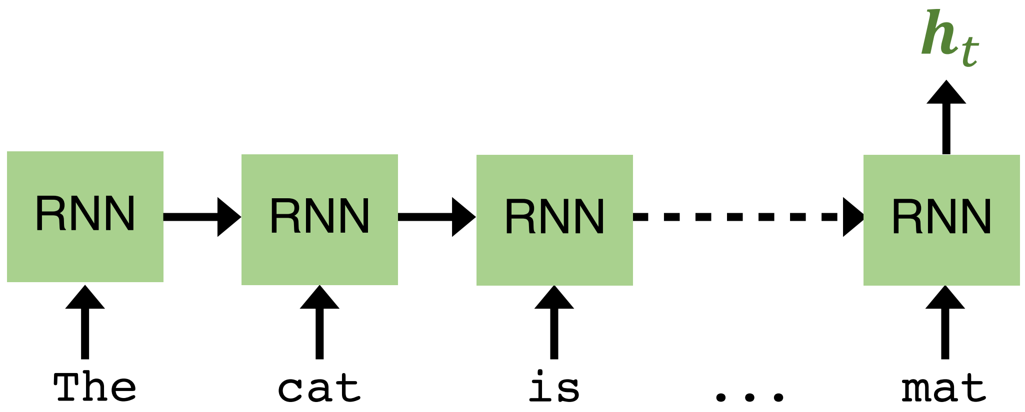


$$V \in \mathbb{R}^{1 \times d}$$

- No spatial information
- Highly compressed

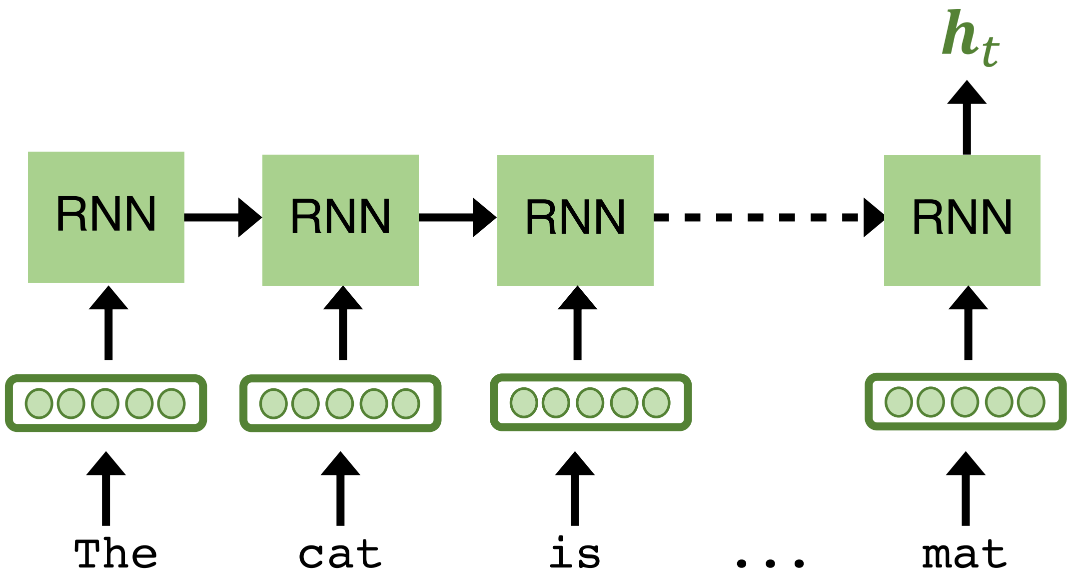
Modeling text

Encoding text



Modeling text

Encoding text



Word embedding are used

These word embeddings can be
- Initialized randomly and trained for a specific task
- Pretrained and frozen
- Pretrained and fine-tuned for a specific task

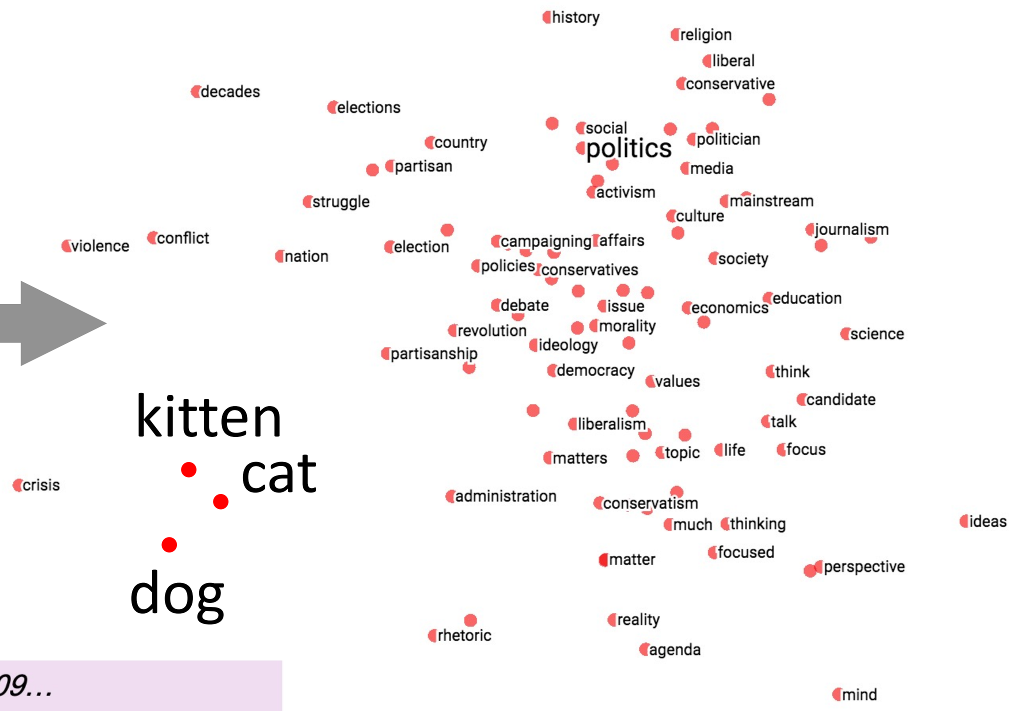
Pretraining can take advantage of huge amount of text-only data

How to pretrain these embeddings?

Text Embeddings

Project words onto a continuous vector space

Similar words closer to each other



...government debt problems turning into ***banking*** *crises as happened in 2009...*

...saying that Europe needs unified ***banking*** *regulation to replace the hodgepodge...*

...India has just given its ***banking*** *system a shot in the arm...*

These context words will represent ***banking***

Distributional hypothesis

"words that occur in similar contexts tend to have similar meanings"

How are these embeddings learned?

Simplify context to small window of adjacent words

Learn to fill in the blank

C1: A bottle of ____ is on the table.

C2: Everybody likes ____.

C3: Don't have ____ before you drive.

C4: We make ____ out of corn.

	C1	C2	C3	C4
tejuino	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0

Language modeling

How are these embeddings learned?

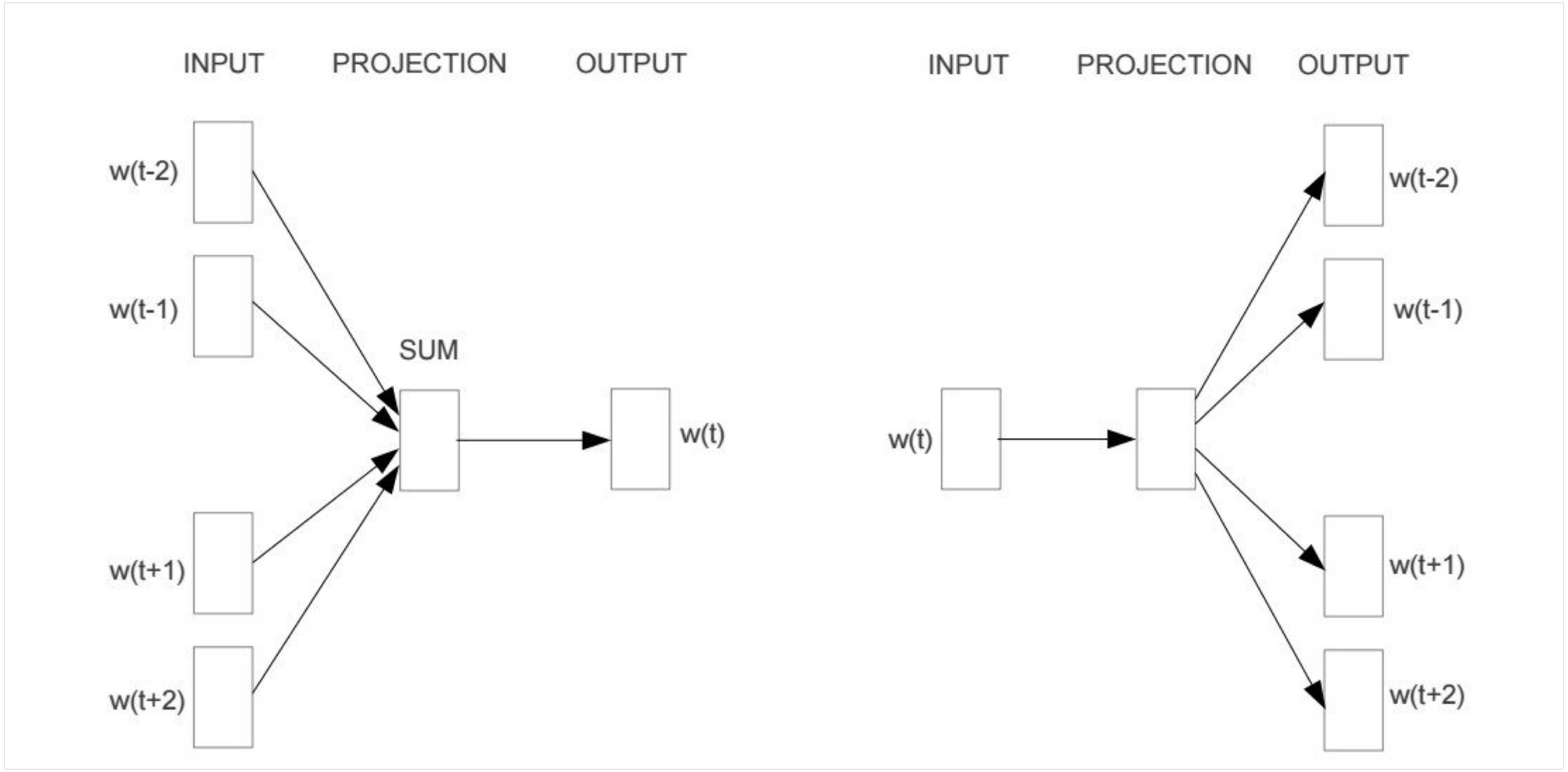
- Represent each word as a vector
- Train classifier to predict word using context words.
- During training, the word vector is updated so that it is possible to predict the center word using the context words.

Simplify context to small window of adjacent words

	C1	C2	C3	C4
tejuino	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0

Word2Vec

Predict center word from context words   Predict context words from center word



Continuous Bag of Words (CBOW)

Skip-grams

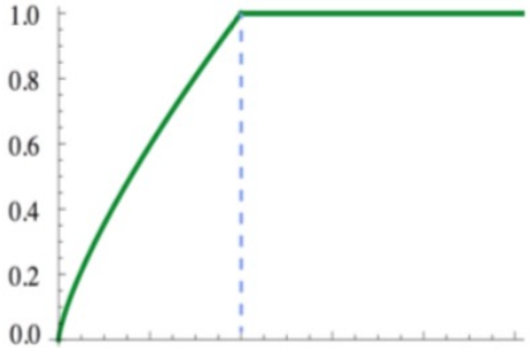
GloVe

- Let's take the global co-occurrence statistics: $X_{i,j}$
- Try to learn word vectors to predict the co-occurence counts (using L2 loss)
- Function f to weight loss by frequency of words (from 0 to 1)

$$J = \sum_{i,j=1}^{|V|} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b_j} - \log X_{ij})^2$$

- Final word vector: $w_i + \tilde{w}_j$
- Training faster
- Scalable to very large corpora

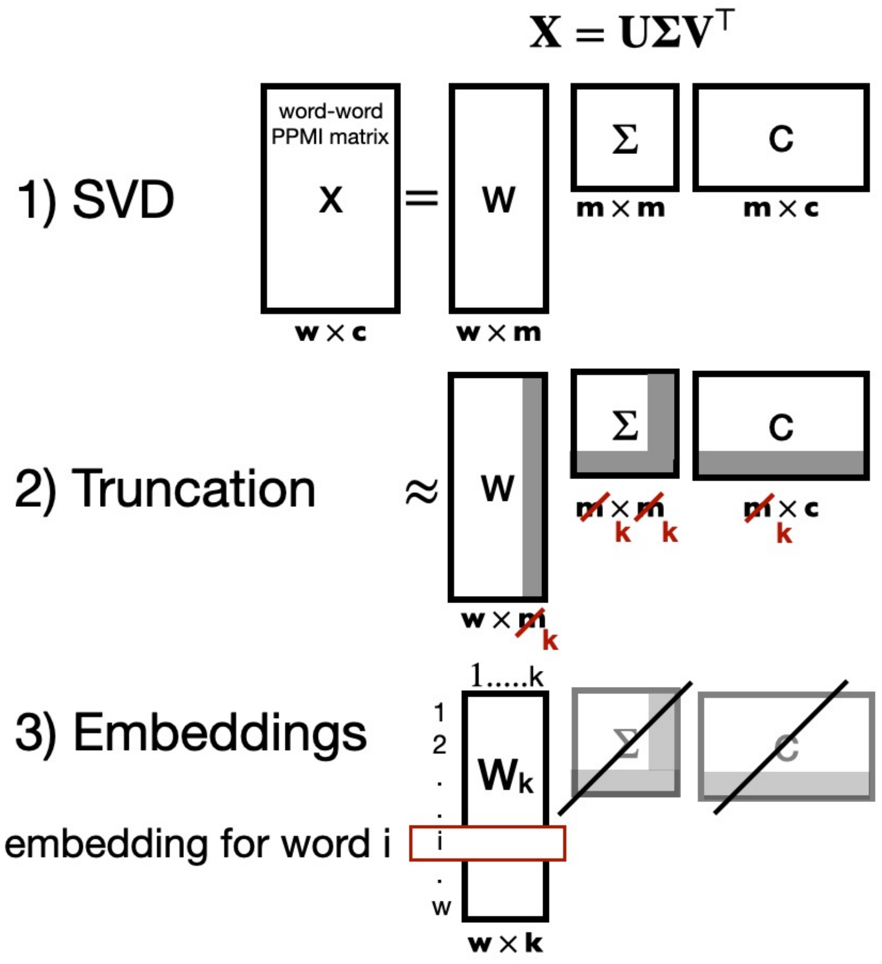
$f \sim$



(Pennington et al, 2014): GloVe: Global Vectors for Word Representation

Factorizing co-occurrence matrix

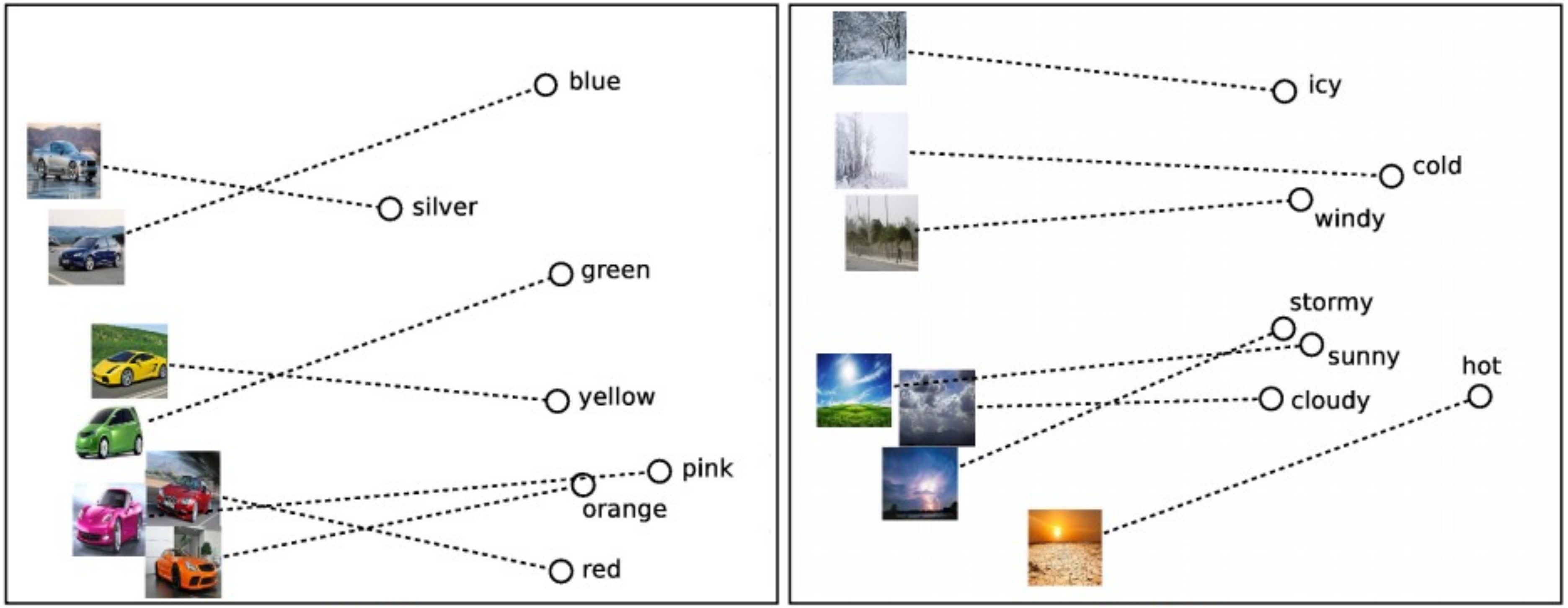
- Obtaining word embeddings via factorization



- Learned word embeddings with word2vec and glove have been shown to be related to factorizing shifted versions of the co-occurrence matrix

Learning multimodal representations

Multimodal Embeddings



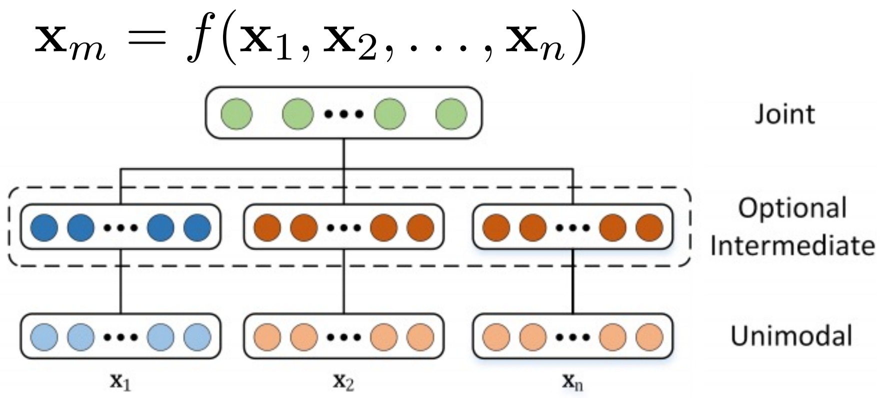
(a) Colors (b) Weather

Figure 5: PCA projection of the 300-dimensional word and image representations for (a) cars and colors and (b) weather and temperature.

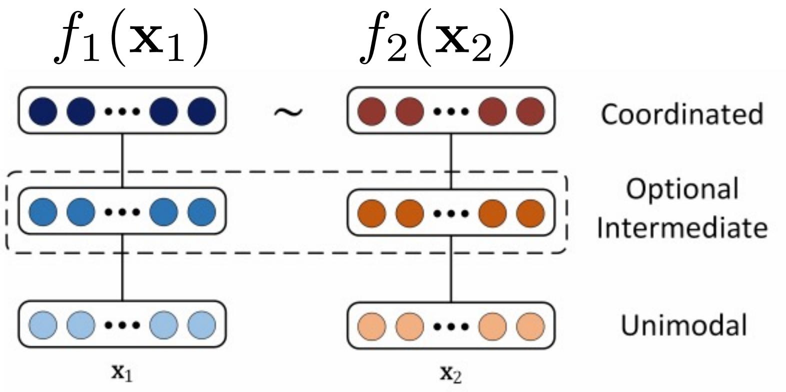
"Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models"
[Kiros, Salakhutdinov, Zemel TACL 2015]

Multimodal representations

- Joint (fused) representations
  - Single combined representation space
  - Early fusion
  - Can be learned supervised or unsupervised



- Coordinated representations
  - Similarity-based methods (e.g. cosine distance)
  - Structure constraints (e.g. orthogonality, sparseness)
  - Examples: CCA, joint embedding



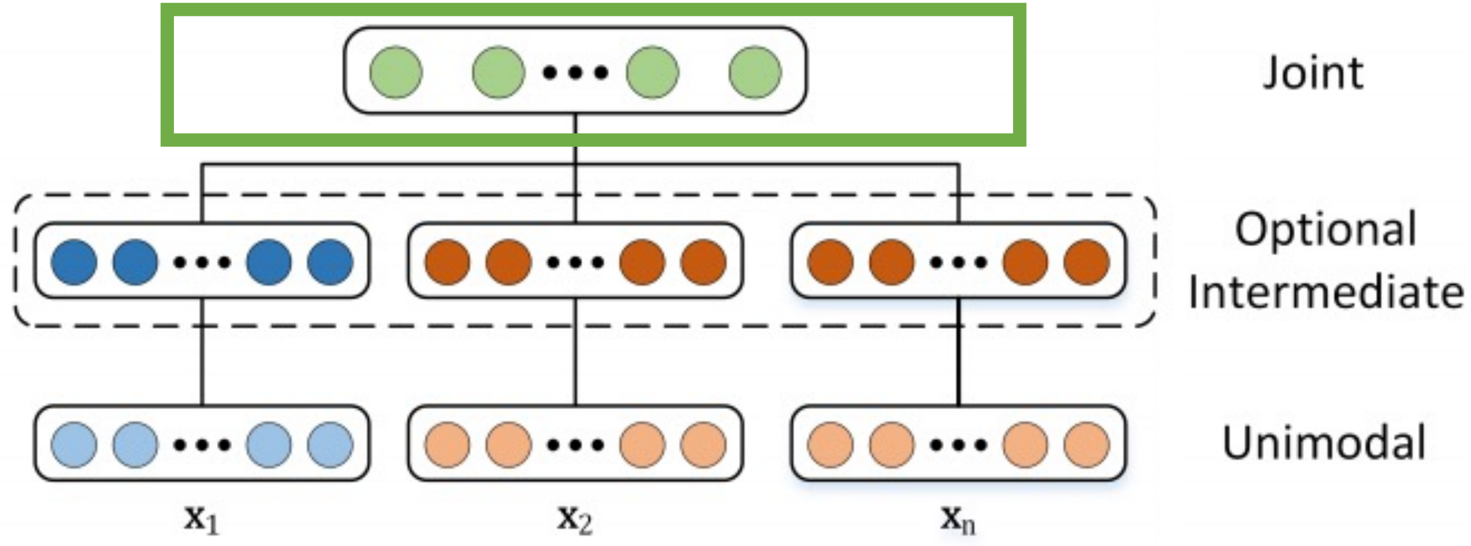
- Representations can be trained end-to-end for a task

Adapted from slide by: Louis-Philippe Morency

Joint representation

- Simplest version: modality concatenation (early fusion)
- More complex: Deep multimodal autoencoders

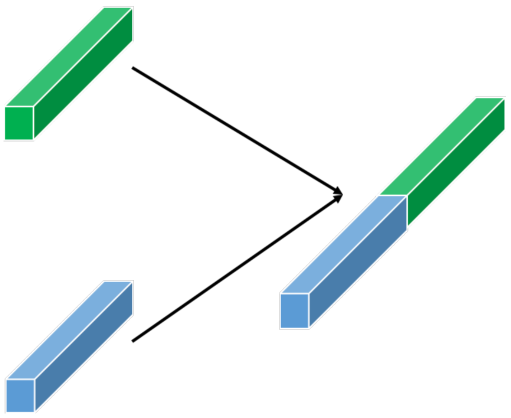
$$\mathbf{x}_m = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$$



Joint representation: Early fusion

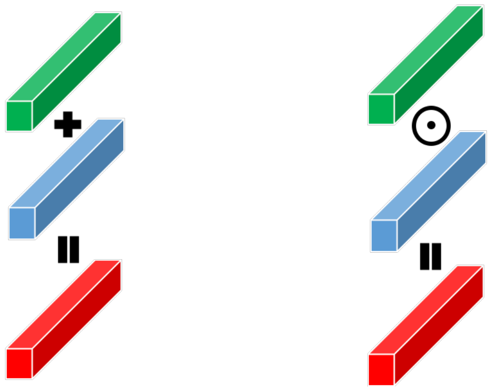
Fusion of features / representation

Concatenation

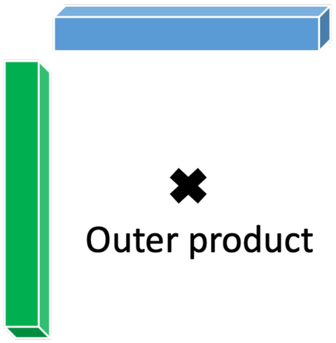


Element wise

Sum Product



Bilinear Pooling



$$z = W\,[x \otimes q]$$

3000 2048 2048

12.5 billion !!!

Image credit: Qi Wu

All elements can interact.
More flexible, but lots of weights!

Adapted from slide by: Stefan Lee

Joint representation: Early fusion

Fusion of features / representation

Bilinear Pooling

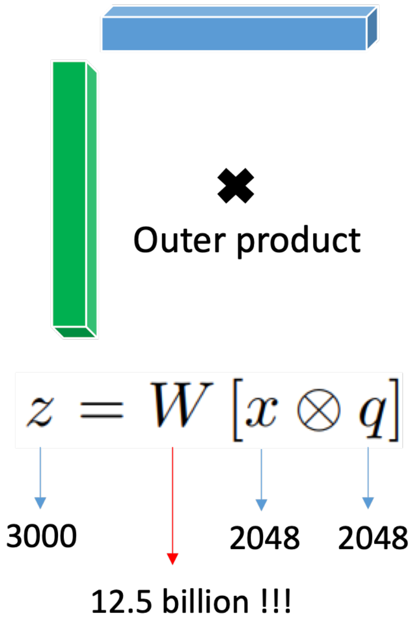


Image credit: Qi Wu

Low rank approximations

All elements can interact.
More flexible, but lots of weights!

Adapted from slide by: Stefan Lee

Joint representation: Early fusion

Compact Bilinear Pooling

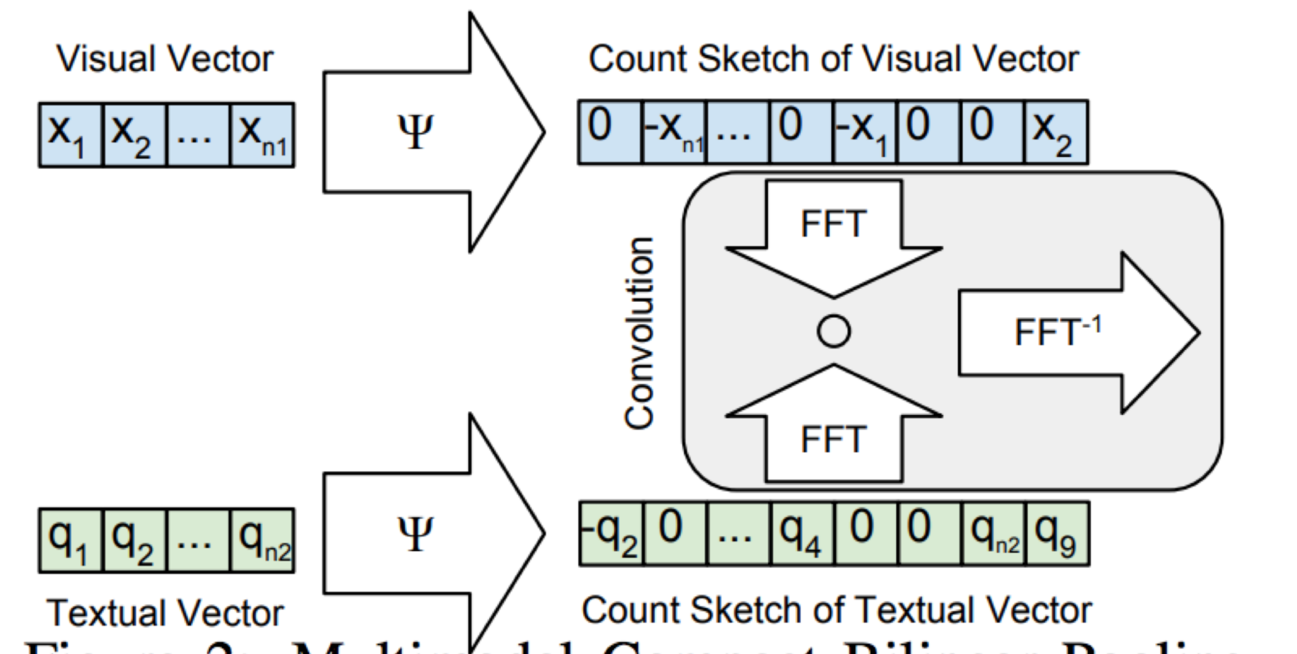


Figure 2: Multimodal Compact Bilinear Pooling

Project outer product to a lower dimensional space

Avoid direct computation of outer product

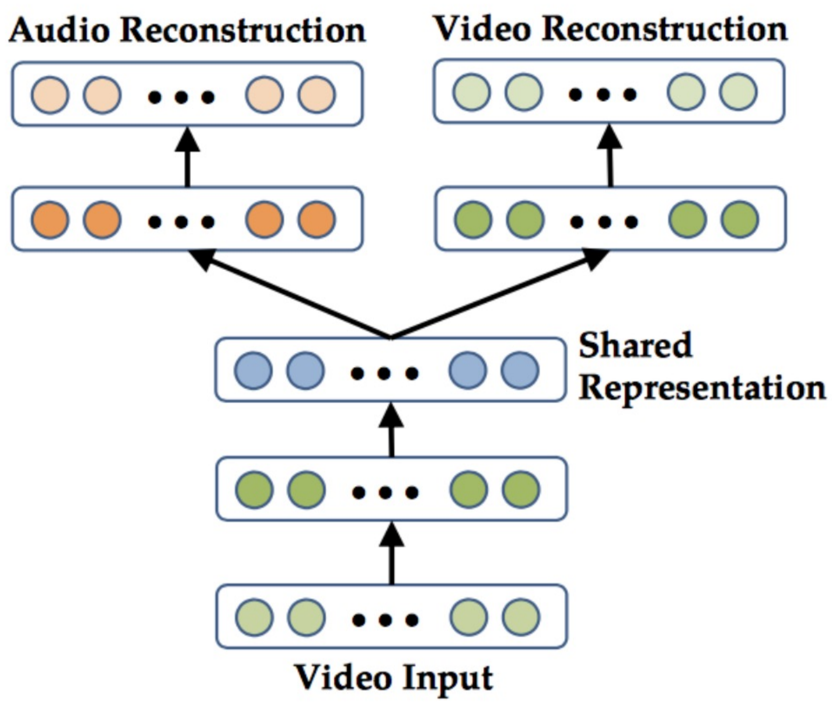
Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding https://arxiv.org/pdf/1606.01847.pdf

Slide credit: Stefan Lee

Joint representation: Autoencoders

Deep Multimodal Autoencoders

- Useful for conditioning on one modality at test time
- Can be regarded as a form of regularization

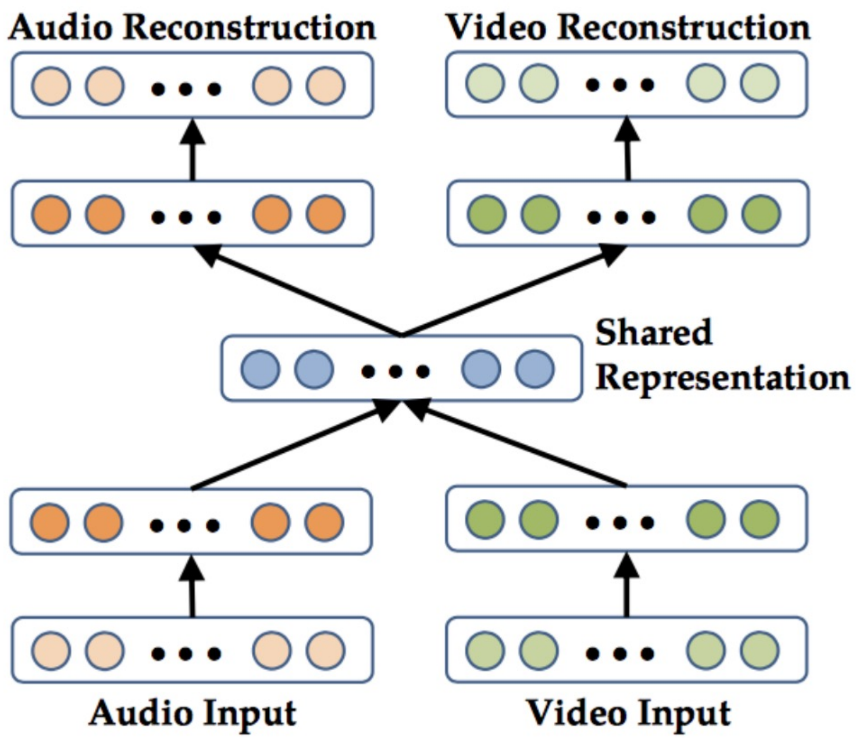


Multimodal deep learning
[Ngiam et al, ICML 2011]

Joint representation: Autoencoders

Deep Multimodal Autoencoders

- Each modality can be pre-trained
  - using denoising autoencoder
- To train the model, reconstruct both modalities using
  - both Audio & Video
  - just Audio
  - just Video



Multimodal deep learning
[Ngiam et al, ICML 2011]

Correlated representations

Canonical correlation analysis (CCA)

- Find representations $f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)$ for each view that maximize correlation:

$$\mathbf{corr}(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)) = \frac{\mathbf{cov}(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2))}{\sqrt{\mathbf{var}(f_1(\mathbf{x}_1)) \cdot \mathbf{var}(f_2(\mathbf{x}_2))}}$$

Joint Embeddings

- Minimize distance between ground truth pairs of samples

$$\min_{f_1, f_2} D\left(f_1(\mathbf{x}_1^{(i)}), f_2(\mathbf{x}_2^{(i)})\right)$$

Canonical Correlation Analysis (CCA)

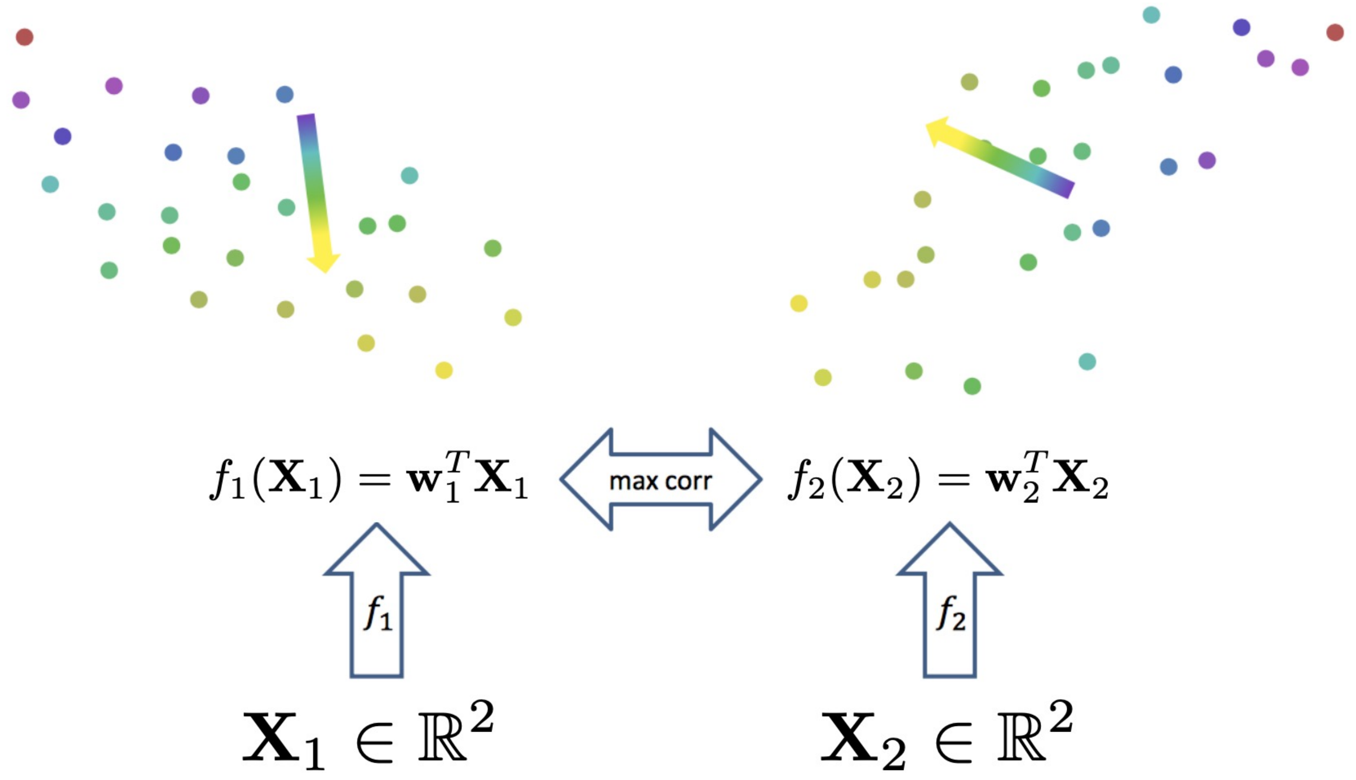
- Goal: Find representations $f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)$ for each view that maximize correlation:

$$\mathbf{corr}(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)) = \frac{\mathbf{cov}(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2))}{\sqrt{\mathbf{var}(f_1(\mathbf{x}_1)) \cdot \mathbf{var}(f_2(\mathbf{x}_2))}}$$

- Finding correlated representations can be useful for
  - Gaining insights into the data
  - Detecting of asynchrony in test data
  - Removing noise uncorrelated across views
  - Translation or retrieval across views

Linear CCA

- Projections of representation



Two views of each instance have the same color

Slide credit: Andrew, Arora, Bilmes, Livescu

Linear CCA

- Classical technique to find linear correlated representations

$$f_1(\mathbf{x}_1) = \mathbf{W}_1^T \mathbf{x}_1 \qquad \text{where} \qquad \mathbf{W}_1 \in \mathbb{R}^{d_1 \times k}$$

$$f_2(\mathbf{x}_2) = \mathbf{W}_2^T \mathbf{x}_2 \qquad\qquad\qquad \mathbf{W}_2 \in \mathbb{R}^{d_2 \times k}$$

- Select values for the first columns $(\mathbf{w}_{1,:1}, \mathbf{w}_{2,:1})$ of the matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to maximize the **correlation of the projections**:

$$(\mathbf{w}_{1,:1}, \mathbf{w}_{2,:1}) = \arg\max \mathbf{corr}(\mathbf{w}_{1,:1}^T \mathbf{X}_1, \mathbf{w}_{2,:1}^T \mathbf{X}_2)$$

- Subsequent pairs are constrained to be **uncorrelated with previous components** (i.e., for $j < i$)

$$\mathbf{corr}(\mathbf{w}_{1,:i}^T \mathbf{X}_1, \mathbf{w}_{1,:j}^T \mathbf{X}_1) = \mathbf{corr}(\mathbf{w}_{2,:i}^T \mathbf{X}_2, \mathbf{w}_{2,:j}^T \mathbf{X}_2) = 0$$

Slide credit: Andrew, Arora, Bilmes, Livescu

Linear CCA

1. Estimate **covariance matrix** with regularization:

$$\Sigma_{11} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_1^{(i)} - \bar{\mathbf{x}}_1)(\mathbf{x}_1^{(i)} - \bar{\mathbf{x}}_1)^T + r_1\mathbf{I} \qquad \Sigma_{12} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_1^{(i)} - \bar{\mathbf{x}}_1)(\mathbf{x}_2^{(i)} - \bar{\mathbf{x}}_2)^T$$

$$\Sigma_{12} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_1^{(i)} - \bar{\mathbf{x}}_1)(\mathbf{x}_2^{(i)} - \bar{\mathbf{x}}_2)^T \qquad \Sigma_{22} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_2^{(i)} - \bar{\mathbf{x}}_2)(\mathbf{x}_2^{(i)} - \bar{\mathbf{x}}_2)^T + r_2\mathbf{I}$$

2. Form **normalized covariance** matrix: $\mathbf{T} = \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2}$ and its singular value decomposition $\mathbf{T} = \mathbf{U}\mathbf{D}\mathbf{V}^T$

3. **Total correlation** at k is $\sum_{i=1}^{k} D_{ii}$

4. The optimal projection matrices are: $\mathbf{W}_1^* = \Sigma_{11}^{-1/2}\mathbf{U}_k$

$$\mathbf{W}_2^* = \Sigma_{11}^{-1/2}\mathbf{V}_k$$

where $\mathbf{U}_k$ is the first k columns of $\mathbf{U}$.

Slide credit: Andrew, Arora, Bilmes, Livescu

Kernel CCA

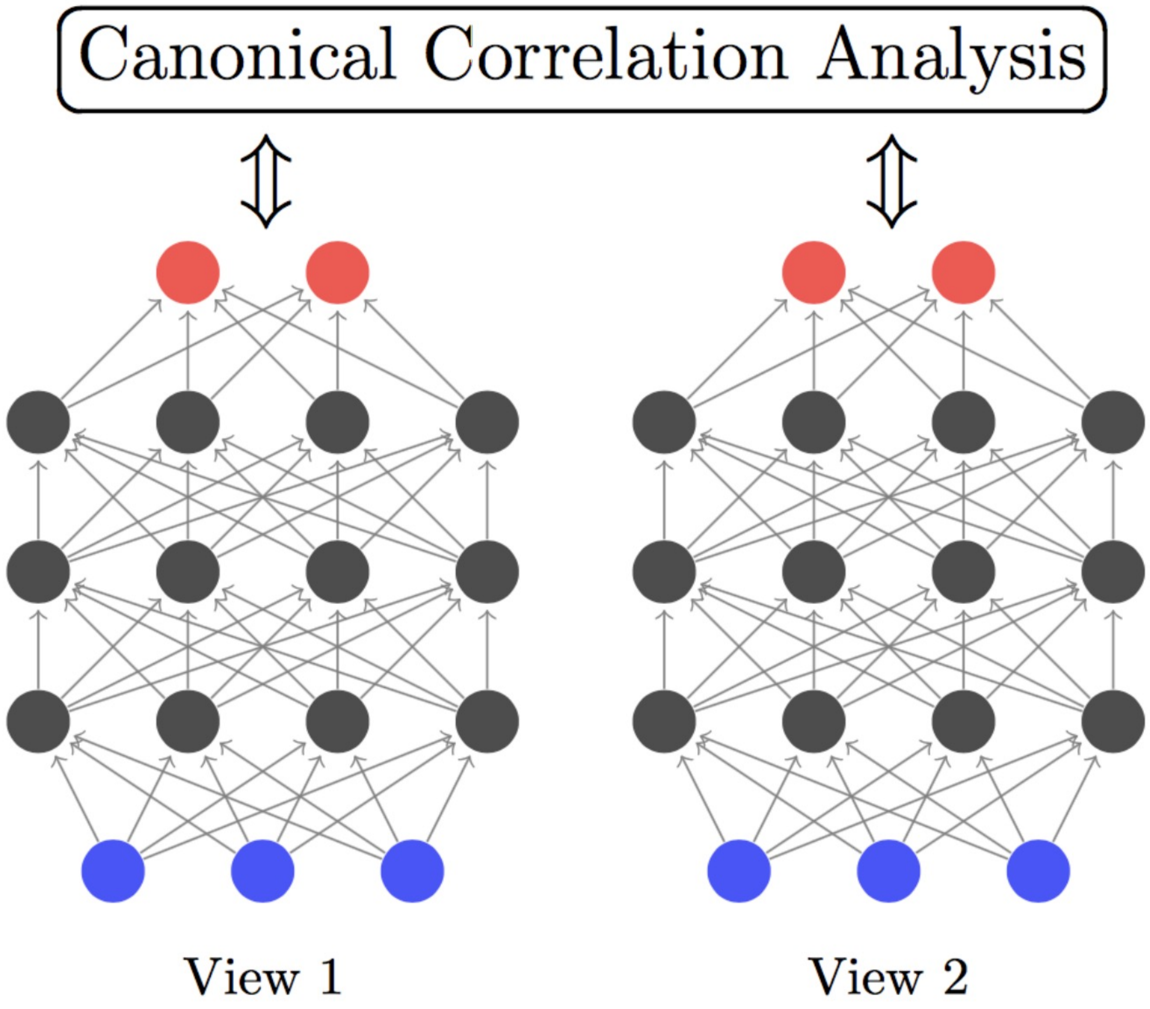
Use non-linear functions for $f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)$

- Learns functions from any reproducing kernel Hilbert space
- May use different kernels for each view
- Using RBF (Gaussian) kernel in KCCA is akin to finding sets of instances that form clusters in both views

- Pros:
  - Allow for non-linear functions
  - Can produce more highly correlated representations
- Cons:
  - KCCA is slower to train
  - KCCA model is more difficult to interpret
  - Training set need to be stored and referenced at test time

Deep CCA

- Use neural network to represent $f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)$
- Can be trained end-to-end for a task



Compared with KCCA

- Training set can be disregarded once the model is learned
- Computational speed at test time is fast

Slide credit: Andrew, Arora, Bilmes, Livescu

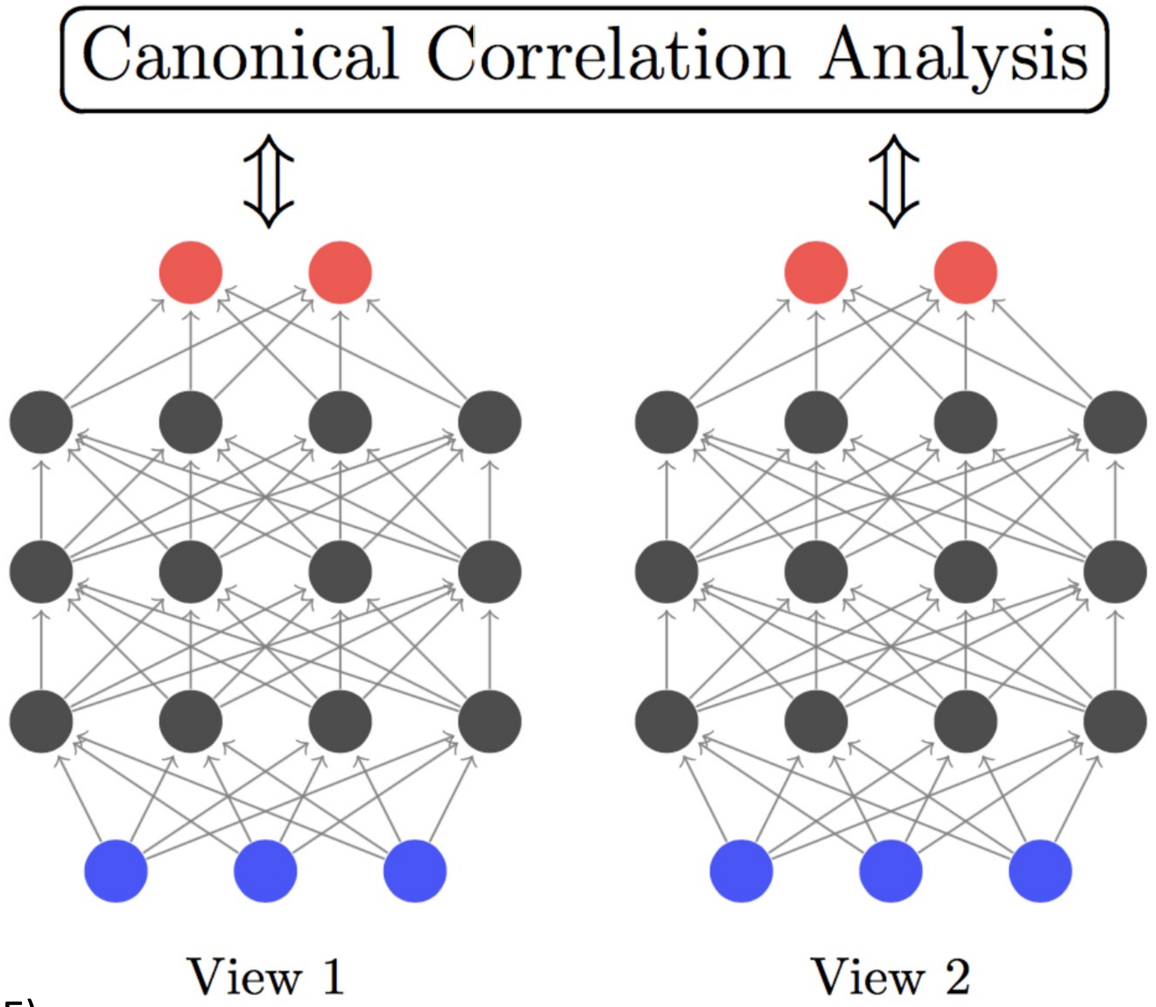
Deep CCA

Training a Deep CCA model:

1. **Pretrain** the layers of **each side** individually

2. **Jointly fine-tune** all parameters to maximize the total correlation of the output layers. Requires computing correlation gradient:
   - Forward propagate activations on both sides.
   - Compute correlation and its gradient w.r.t. output layers.
   - Backpropagate gradient on both sides.

Correlation is a population objective, so instead of one instance (or minibatch) training, requires L-BFGS second-order method (with full-batch)



Extensions: Deep canonically correlated autoencoders (DCCAE)

Slide credit: Andrew, Arora, Bilmes, Livescu

Correlated representations

Canonical correlation analysis (CCA)

- Find representations $f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)$ for each view that maximize correlation:

$$\mathbf{corr}(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2)) = \frac{\mathbf{cov}(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2))}{\sqrt{\mathbf{var}(f_1(\mathbf{x}_1)) \cdot \mathbf{var}(f_2(\mathbf{x}_2))}}$$

Joint Embeddings

- Minimize distance between ground truth pairs of samples
  (or maximize similarity)

$$\min_{f_1, f_2} D\left(f_1(\mathbf{x}_1^{(i)}), f_2(\mathbf{x}_2^{(i)})\right)$$

Discriminative Embeddings

Images and **class labels** are embedded into the same space

Image Embedding

$$\Psi(I_i) = \mathbf{W} \cdot CNN(I_i; \mathbf{\Theta}) : \mathbb{R}^D \to \mathbb{R}^d$$

Label Embedding

$$\Psi_L(word_i) = \mathbf{u}_i : \{1, ..., L\} \to \mathbb{R}^d$$

Distance or Similarity in Embedding Space

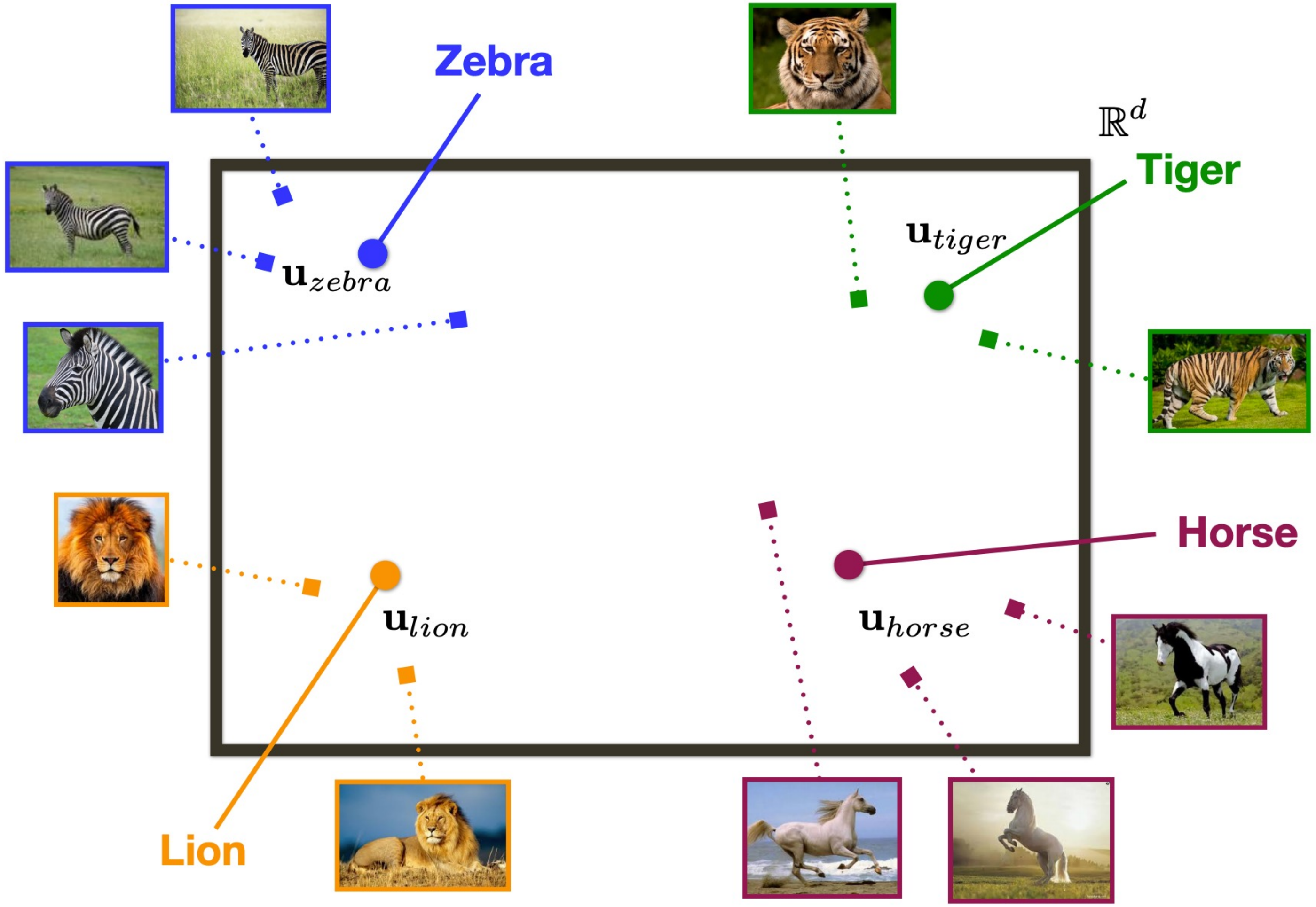
Can use different distances / similarities

Euclidean (L2) distance

$$D(\mathbf{u}, \mathbf{u}') = \|\mathbf{u} - \mathbf{u}'\|_2^2$$

Cosine similarity

$$S(\mathbf{u}, \mathbf{u}') = \frac{\mathbf{u}}{\|\mathbf{u}\|} \cdot \frac{\mathbf{u}'}{\|\mathbf{u}'\|}$$



Adapted from slide by: Leonid Sigal

Discriminative Embeddings

Train network to minimize distance / maximize similarity!

Image Embedding

$$\Psi(I_i) = \mathbf{W} \cdot CNN(I_i; \mathbf{\Theta}) \colon \mathbb{R}^D \rightarrow \mathbb{R}^d$$

Label Embedding

$$\Psi_L(word_i) = \mathbf{u}_i : \{1, ..., L\} \rightarrow \mathbb{R}^d$$

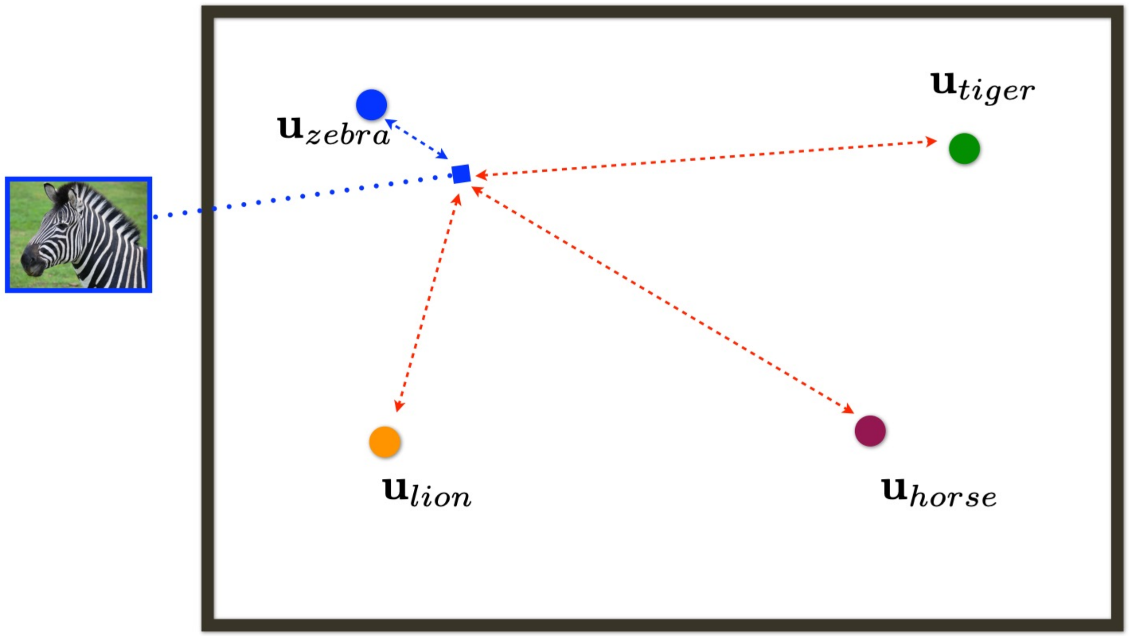
Correct label (more similar)

Other labels (less similar)

$$\mathcal{L}_C(\mathbf{W}, \mathbf{U}, I_i, y_i) = \sum \max\left[0, \alpha - \underbrace{S(\Psi(I_i), \mathbf{u}_{y_i})}_{\text{Correct label}} + \underbrace{S(\Psi(I_i), \mathbf{u}_{y_c})}_{\text{Other labels}}\right]$$

Take care with signs depending if working with similarity or distance

$\mathbb{R}^d$



Similarity in Embedding Space

$$S(\mathbf{u}, \mathbf{u}') = \frac{\mathbf{u}}{||\mathbf{u}||} \cdot \frac{\mathbf{u}'}{||\mathbf{u}'||}$$

Objective Function:

$$\min_{\mathbf{W}, \mathbf{U}} \sum_i^N \mathcal{L}_C(\mathbf{W}, \mathbf{U}, I_i, y_i) + \lambda_1 ||\mathbf{W}||_F^2 + \lambda_2 ||\mathbf{U}||_F^2$$

[Bengio *et al.*,, NIPS'10]

[Weinberger, Chapelle, NIPS'09]

Adapted from slide by: Leonid Sigal

From words to sentences

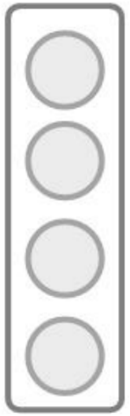
Label Embedding

$\Psi_L(word_i) = \mathbf{u}_i : \{1, ..., L\} \rightarrow \mathbb{R}^d$

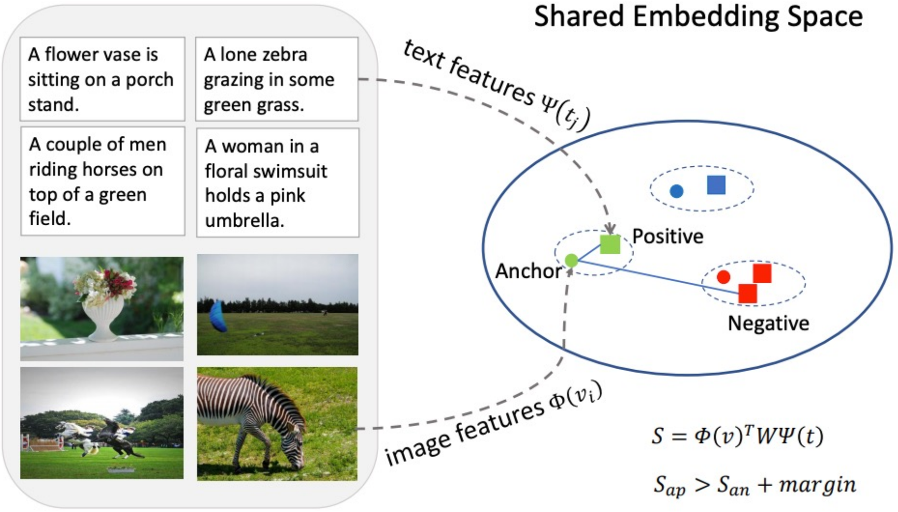
Sentence embedding

$$\Psi_L(w_1, \ldots, w_k)$$

Sequence of words (characters) → Composition Function →



Average
BoW (FCN)
RNN
CNN
Transformers
GraphNN



(i, c) : matching

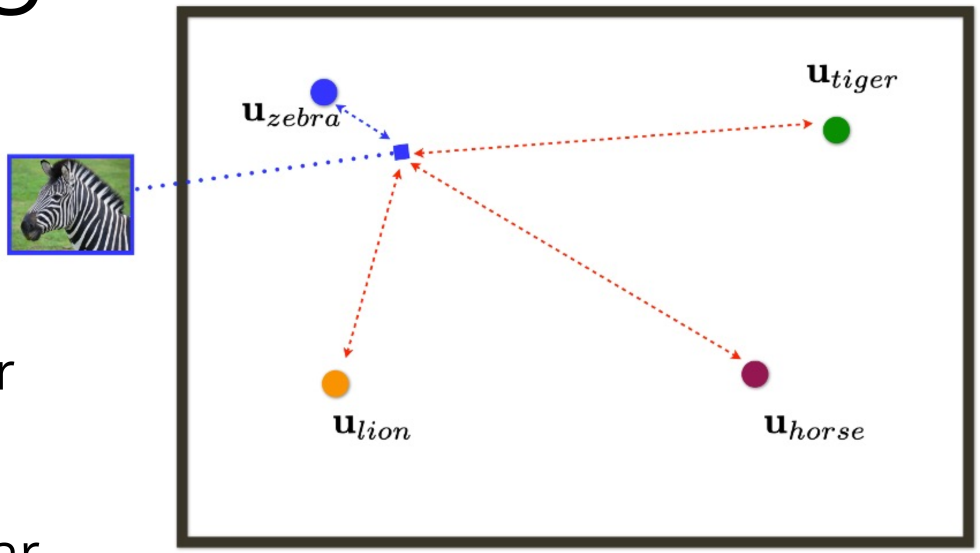
$(\hat{i}, c), (i, \hat{c})$: not matching

Triplet based ranking loss:

$$\ell_{SH}(i,c) = \sum_{\hat{c}}[\alpha - s(i,c) + s(i,\hat{c})]_+ + \sum_{\hat{i}}[\alpha - s(i,c) + s(\hat{i},c)]_+$$

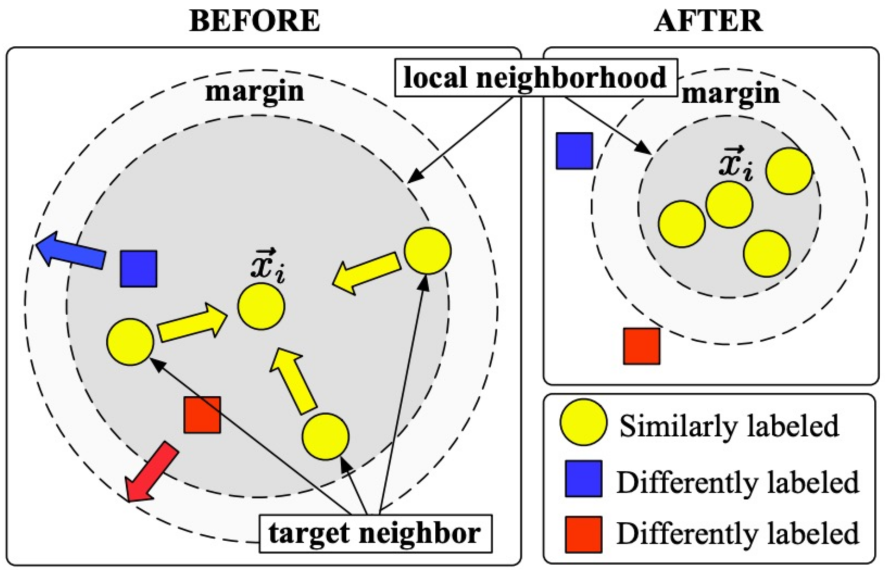
Discrminative Embeddings

- Inputs are mapped into a feature space
- Want the following:
  - pairs that have the same label to have similar features (i.e. be close together in the feature space)
  - pairs that have different labels to be dissimilar (i.e. be be far apart in the feature space)
- Rich literature in this area with
  - different loss functions
  - how to construct positive and negative examples



Contrastive and metric learning

- Metric Learning: Learning distance metric that can separate input with the same label from those with different labels



"Distance Metric Learning for Large Margin Nearest Neighbor Classification"
[Weinberger, Blitzer and Saul, NIPS 2005]

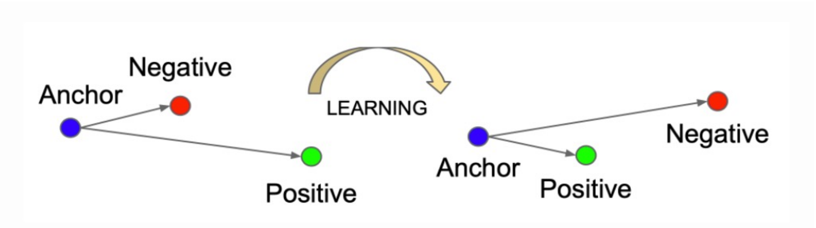
- Contrastive Learning: Learning similarity metric discriminatively

Losses

- Contrastive Loss
  - Proposed for face verification (Chopra et al., 2005)
  - Pairwise ranking loss

$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{1}[y_i = y_j] D(f(\mathbf{x}_i), f(\mathbf{x}_j)) + \mathbb{1}[y_i \neq y_j] \max\left(0, \epsilon - D(f(\mathbf{x}_i), f(\mathbf{x}_j))\right)$$

- Triplet Loss
  - Proposed in FaceNet (Schroff et al., 2015)
  - Select anchor with positive and negative



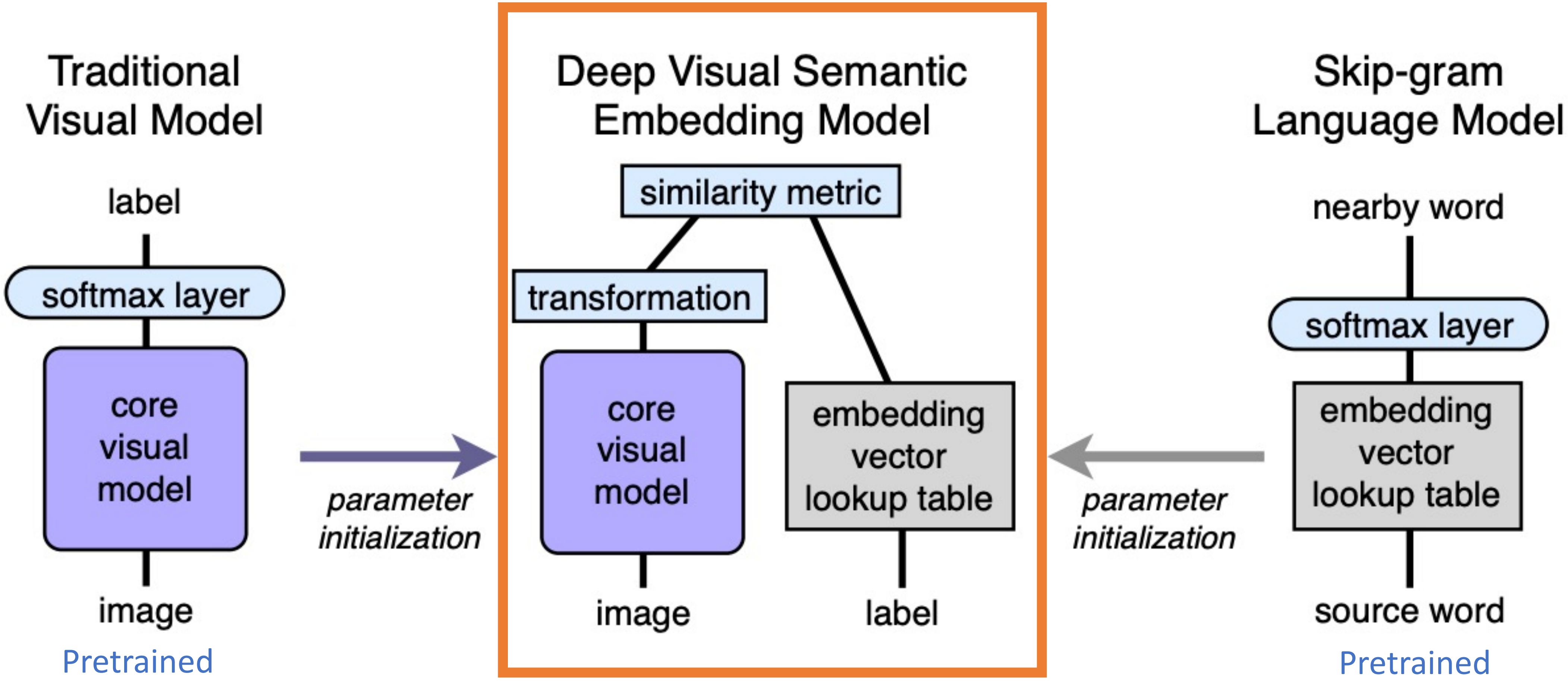
$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{x \in \mathcal{X}} \max\left(0, \epsilon + D(f(\mathbf{x}), f(\mathbf{x}^+)) - D(f(\mathbf{x}), f(\mathbf{x}^-))\right)$$

(a) Contrastive embedding (b) Triplet embedding

Figure from "Deep Metric Learning via Lifted Structured Feature Embedding"
[Song et al, CVPR 2016]

Using triplet loss in multimodal embeddings

$$loss(image, label) = \sum_{j \neq label} \max[0, margin - \vec{t}_{label} M \vec{v}(image) + \vec{t}_j M \vec{v}(image)]$$



"DeViSE: A Deep Visual-Semantic Embedding Model"
[Frome et al, NIPS 2013]

Training data

- Positive pairs
  - Correctly labeled data: (image, label) or (image, description)
  - Perturb input for data augmentation

- Negative pairs
  - Sample non-matching pairs
    - What kind of negatives to sample?
  - How to efficiently sample?

Going beyond triplets

- Consider all pairs in a batch for efficient in-batch sampling

(N+1) Tuplet

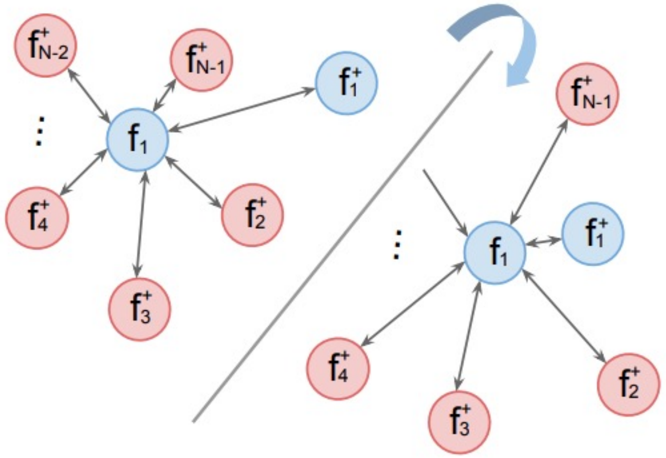


Figure from “Improved Deep Metric Learning with Multi-class N-pair Loss Objective”
[Sohn, NIPS 2016]

Lifted Structured Feature Embedding

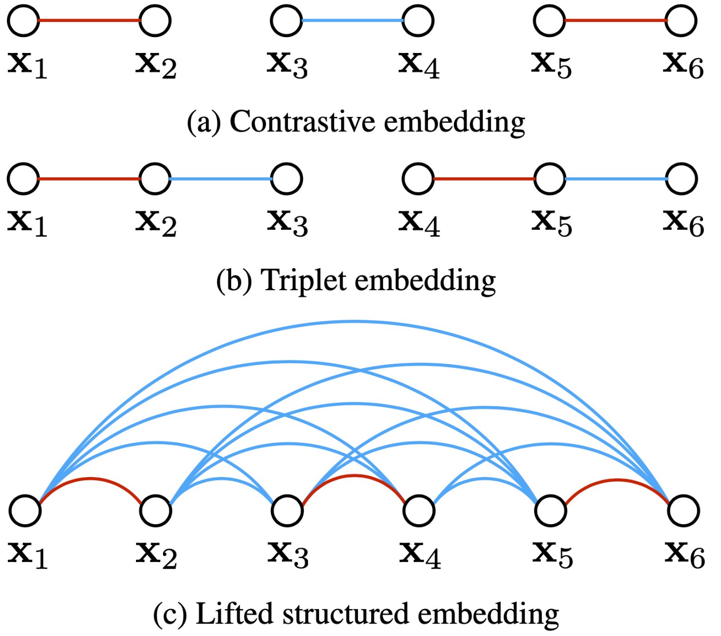


Figure from “Deep Metric Learning via Lifted Structured Feature Embedding”
[Song et al, CVPR 2016]

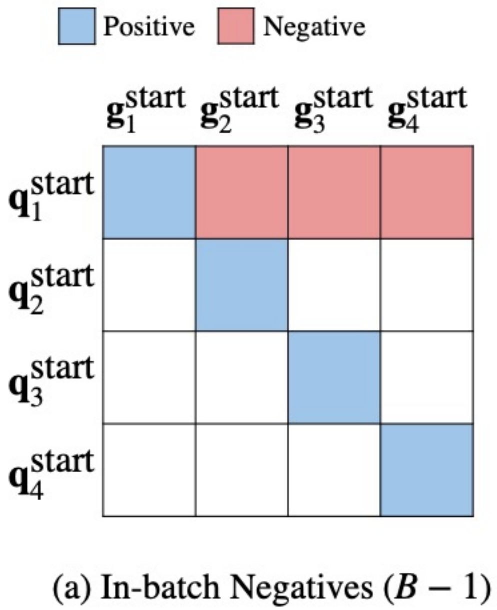
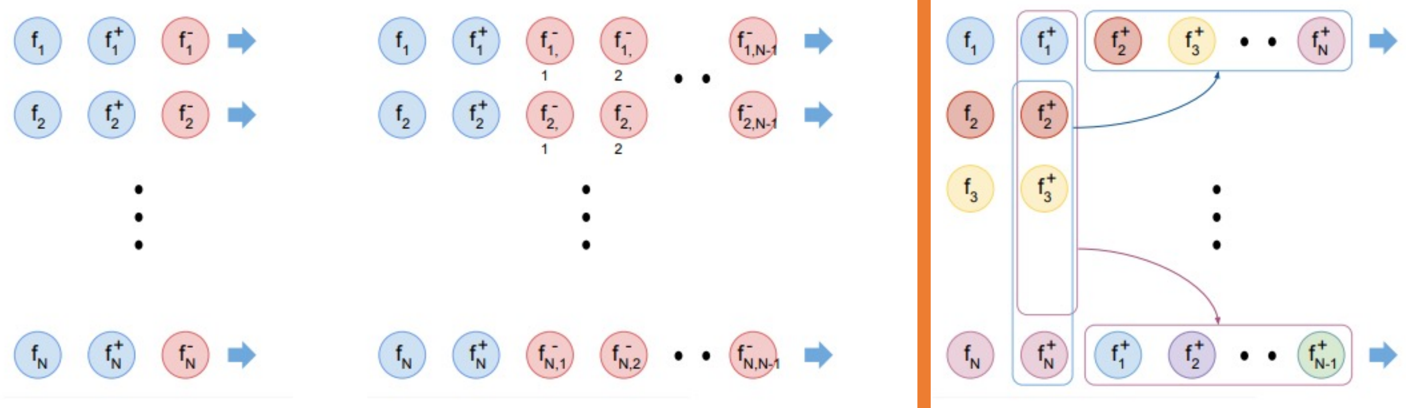


Figure from “Learning Dense Representations of Phrases at Scale”
[Lee et al, ACL 2021]

Contrastive learning as classification

- N-paired Multiclass loss



(a) Triplet loss (b) $(N+1)$-tuplet loss (c) N-pair-mc loss

$$\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right)$$

$$= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}$$

"Improved Deep Metric Learning with Multi-class N-pair Loss Objective"
[Sohn, NIPS 2016]

Contrastive learning as classification

- Noise Contrastive Estimation
  - Train logistic regression classifier to distinguish positive and negative (noise) samples
  - Uses cross-entropy loss
    - With one positive sample and one noise sample (Gutmann & Hyvarinen, 2010)

$$\mathcal{L}_{\mathrm{NCE}} = -\frac{1}{N} \sum_{i=1}^{N} [\log \sigma(\ell_\theta(\mathbf{x}_i)) \log(1 - \sigma(\ell_\theta(\tilde{\mathbf{x}}_i)))]$$

    - With multiple noise samples (InfoNCE, van den Oord et al., 2018)

$$\mathcal{L}_{\mathrm{InfoNCE}} = -\mathbb{E}\Big[\log \frac{f(\mathbf{x}, \mathbf{c})}{\sum_{\mathbf{x}' \in X} f(\mathbf{x}', \mathbf{c})}\Big]$$

Contrastive learning as classification

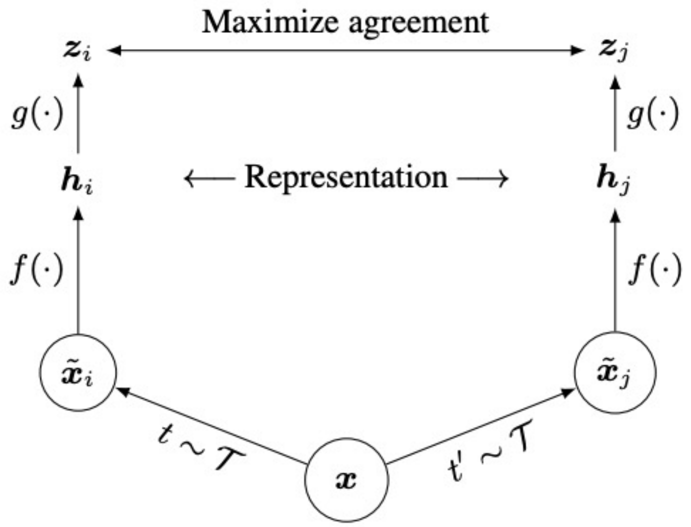
- Temperature Scaled
  - Temperature parameter τ controls how spiky /smooth the distribution is
  - Automatically weights examples by their "hardness"

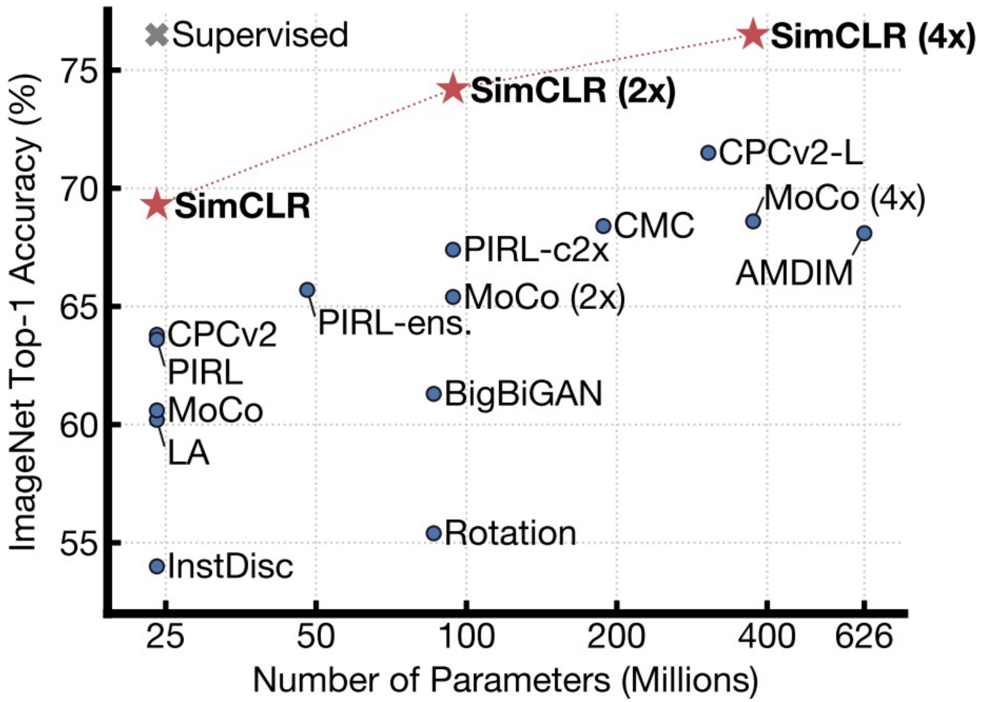
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)} ,$$

"Learning a Similarity Metric Discriminatively, with Application to Face Verification"
[Chopra, Hadsell and LeCun, CVPR 2005]

SimCLR

- Does data augmentation help?
- What loss function to use?
- Effect of batch size and other hyperparameters

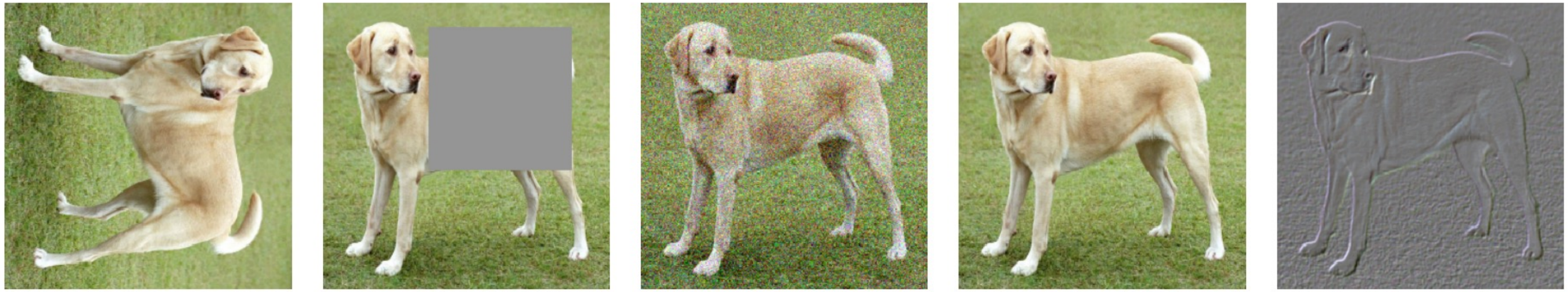




"A Simple Framework for Contrastive Learning of Visual Representations"
[Chen et al., ICML 2020]

Injecting Noise / Data Augmentation

(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)



(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

"A Simple Framework for Contrastive Learning of Visual Representations"
[Chen et al., ICML 2020]

What loss to use?

A Simple Framework for Contrastive Learning of Visual Representations

Name	Negative loss function	Gradient w.r.t. $\boldsymbol{u}$
NT-Xent	$\boldsymbol{u}^T\boldsymbol{v}^+/\tau - \log\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^+,\boldsymbol{v}^-\}}\exp(\boldsymbol{u}^T\boldsymbol{v}/\tau)$	$(1-\frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^+/\tau)}{Z(\boldsymbol{u})})/\tau\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^-}\frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)}{Z(\boldsymbol{u})}/\tau\boldsymbol{v}^-$
NT-Logistic	$\log\sigma(\boldsymbol{u}^T\boldsymbol{v}^+/\tau) + \log\sigma(-\boldsymbol{u}^T\boldsymbol{v}^-/\tau)$	$(\sigma(-\boldsymbol{u}^T\boldsymbol{v}^+/\tau))/\tau\boldsymbol{v}^+ - \sigma(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)/\tau\boldsymbol{v}^-$
Margin Triplet	$-\max(\boldsymbol{u}^T\boldsymbol{v}^- - \boldsymbol{u}^T\boldsymbol{v}^+ + m, 0)$	$\boldsymbol{v}^+ - \boldsymbol{v}^-$ if $\boldsymbol{u}^T\boldsymbol{v}^+ - \boldsymbol{u}^T\boldsymbol{v}^- < m$ else $\mathbf{0}$

Normalized dot product
(cosine similarity)

Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

"A Simple Framework for Contrastive Learning of Visual Representations"
[Chen, ICML 2020]

Effect of batch size and other hyperparameters

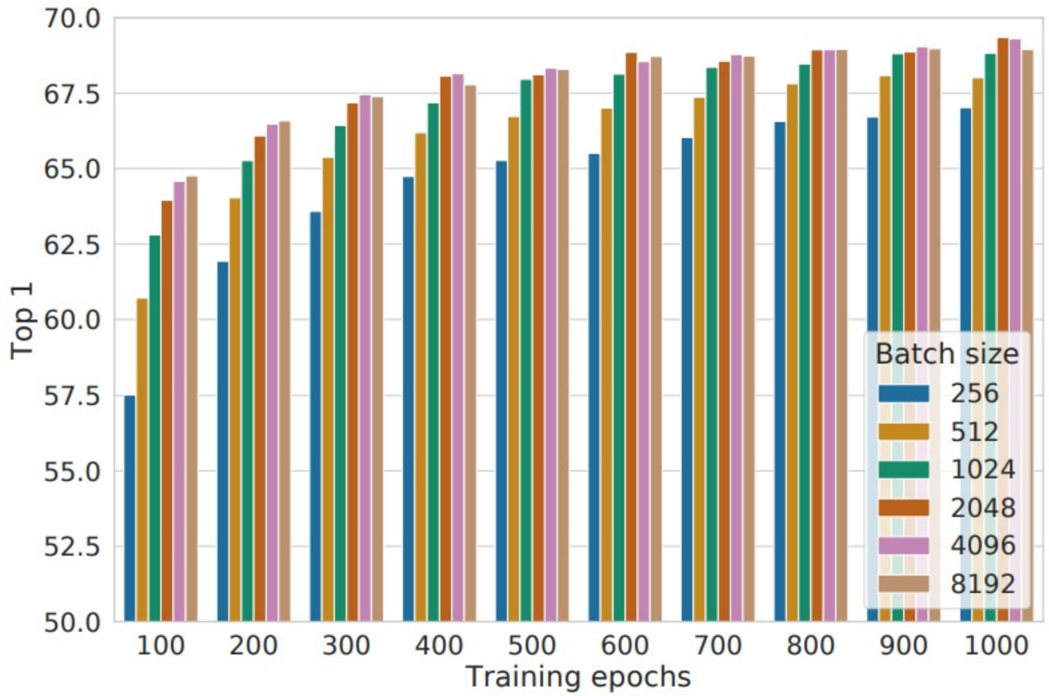


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

ℓ_2 norm?	τ	Entropy	Contrastive acc.	Top 1
Yes	0.05	1.0	90.5	59.7
	0.1	4.5	87.8	64.4
	0.5	8.2	68.2	60.7
	1	8.3	59.1	58.0
No	10	0.5	91.7	57.2
	100	0.5	92.1	57.0

Table 5. Linear evaluation for models trained with different choices of ℓ_2 norm and temperature τ for NT-Xent loss. The contrastive distribution is over 4096 examples.

"A Simple Framework for Contrastive Learning of Visual Representations"
[Chen, ICML 2020]

Applications

Retrieval

- Text to image/video retrieval
- Image/video to text retrieval

MS COCO

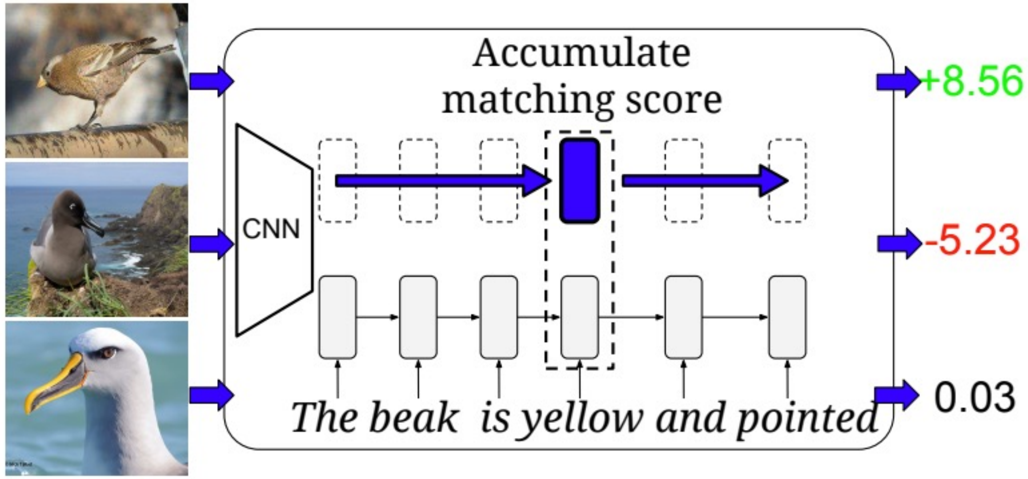
The man at bat readies to swing at the pitch while the umpire looks on.

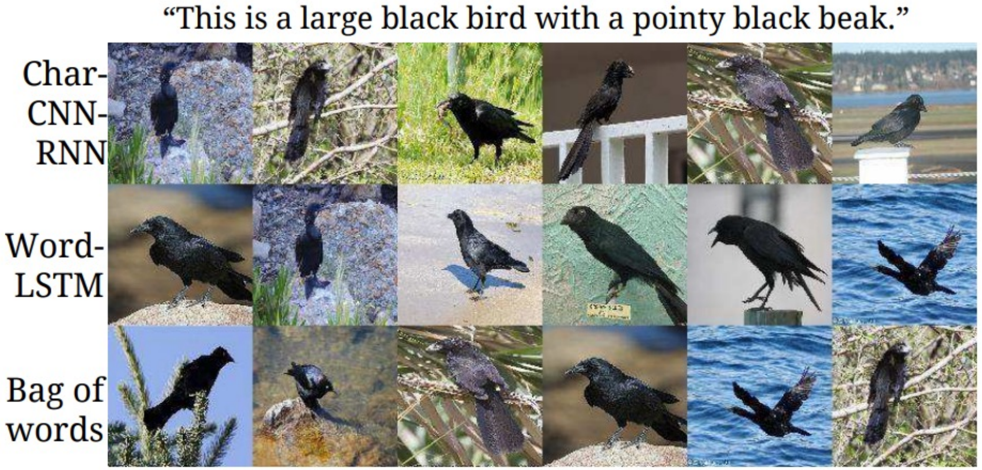
A large bus sitting next to a very tall building.

Flicker 8k, Flicker 30k

- A biker in red rides in the countryside.
- A biker on a dirt path.
- A person rides a bike off the top of a hill and is airborne.
- A person riding a bmx bike on a dirt course.
- The person on the bicycle is wearing red.

Retrieval





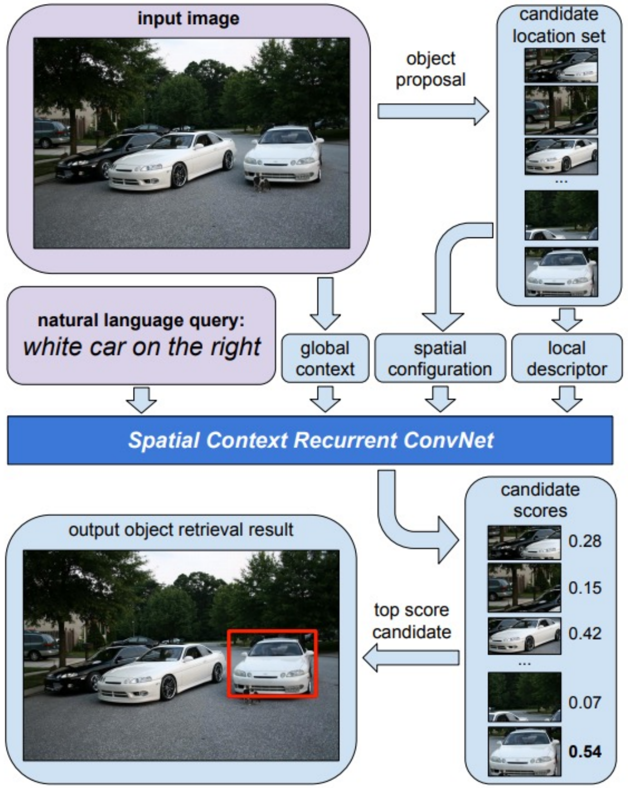
	Top-1 Acc (%)		AP@50 (%)	
Embedding	**DA-SJE**	**DS-SJE**	**DA-SJE**	**DS-SJE**
Attributes	50.9	50.4	20.4	**50.0**
Word2Vec	38.7	38.6	7.5	33.5
Bag-of-Words	43.4	44.1	24.6	39.6
Char CNN	47.2	48.2	2.9	42.7
Char LSTM	22.6	21.6	11.6	22.3
Char CNN-RNN	54.0	54.0	6.9	45.6
Word CNN	50.5	51.0	3.4	43.3
Word LSTM	52.2	53.0	**36.8**	46.8
Word CNN-RNN	**54.3**	**56.8**	4.8	48.7

CUB Birds

"Learning Deep Representations of Fine-Grained Visual Descriptions" (Reed et al, CVPR 2016)

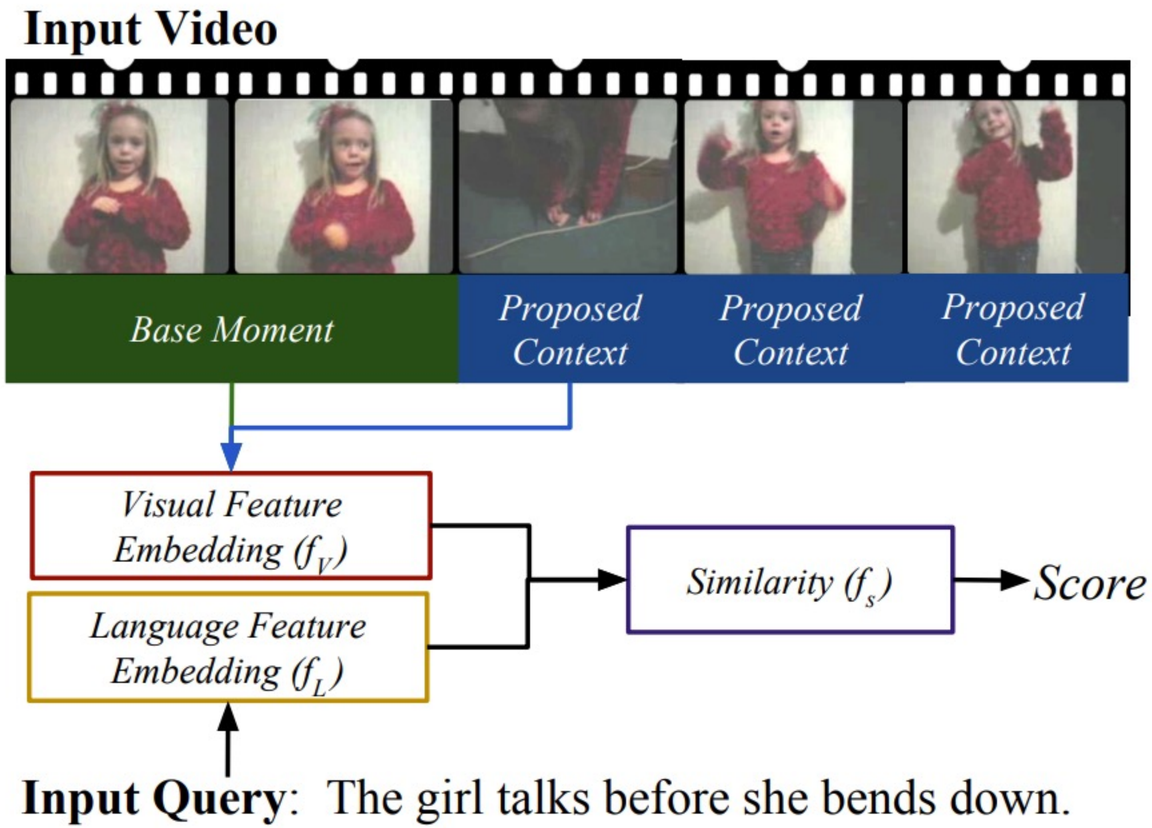
Retrieval

Match image region to language



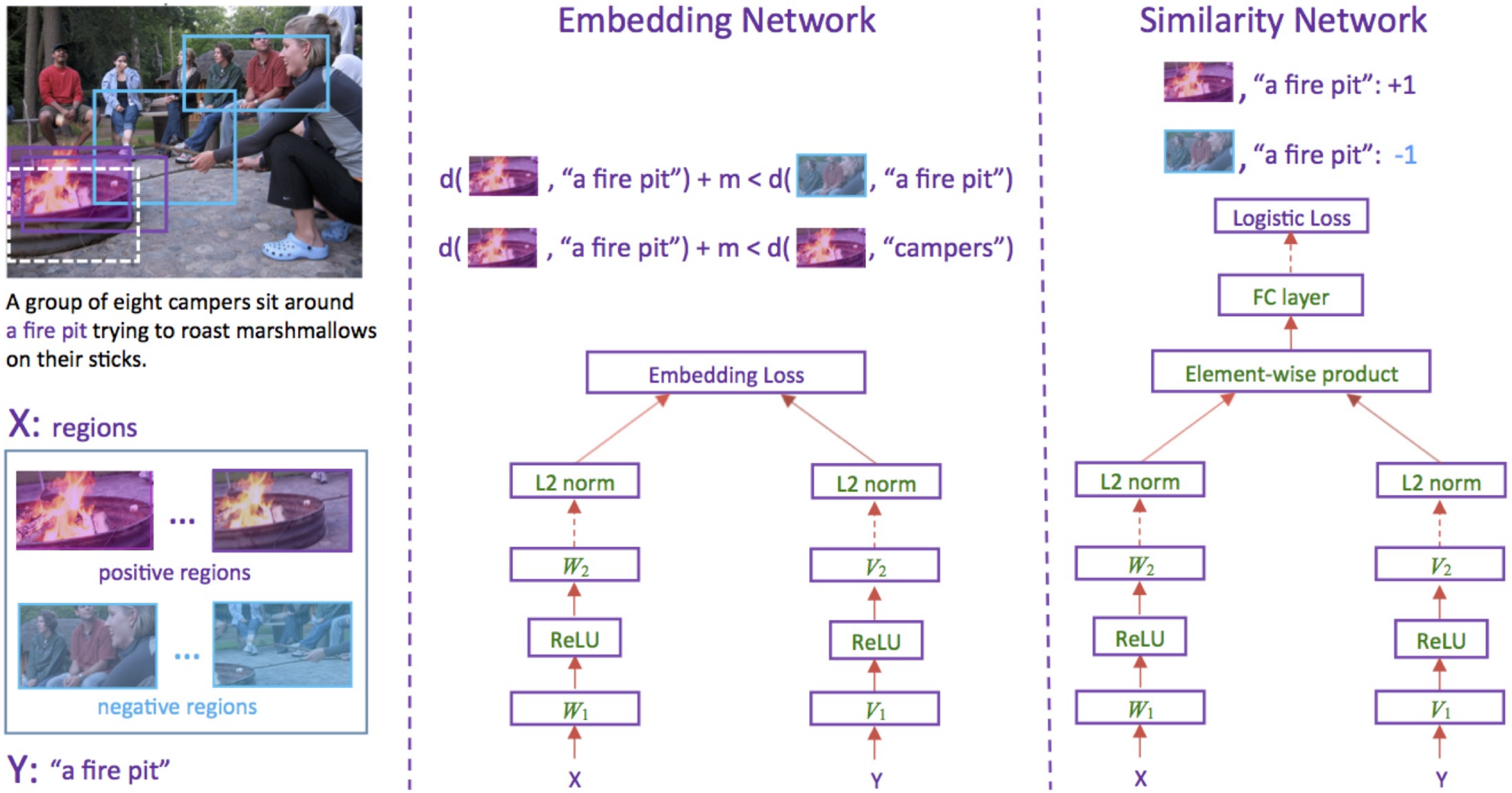
Natural Language Object Retrieval
(Hu et al, CVPR 2016)

Match video frames to language



Localizing moments in video with temporal language
(Hendricks et al, EMNLP, 2018)

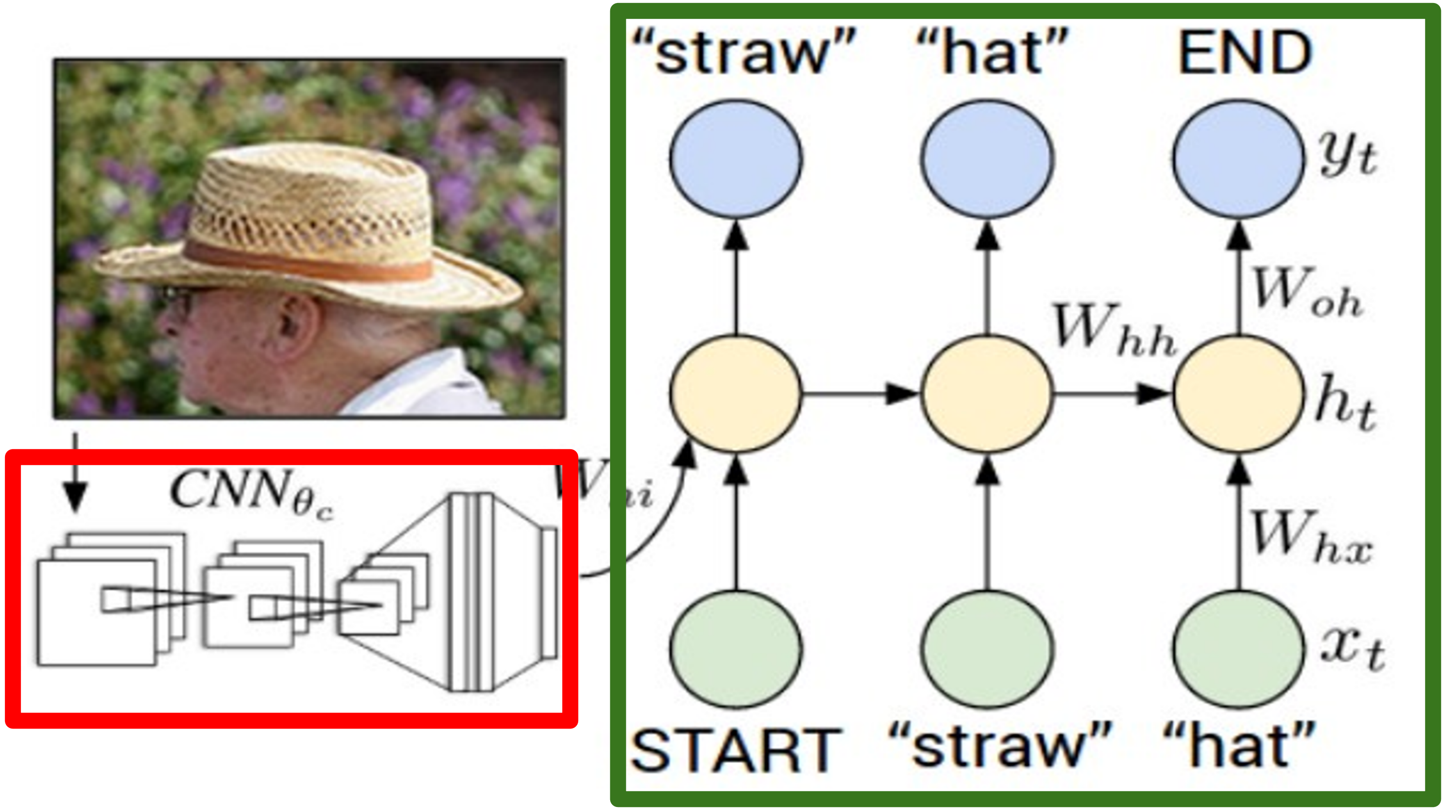
Retrieval: Phrase localization



Learning Two-Branch Neural Networks for Image-Text Matching Tasks
(Wang et al, TPAMI 2018)

Translation (image to text)

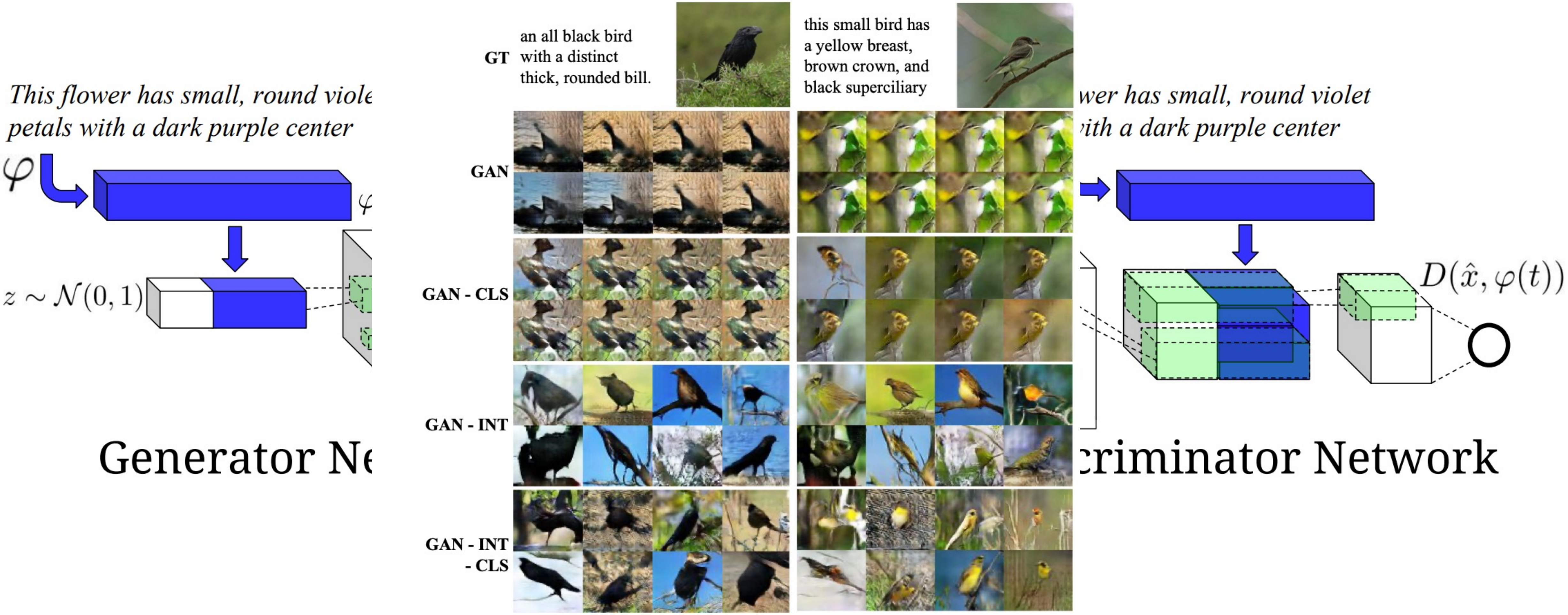
Recurrent Neural Network



Convolutional Neural Network

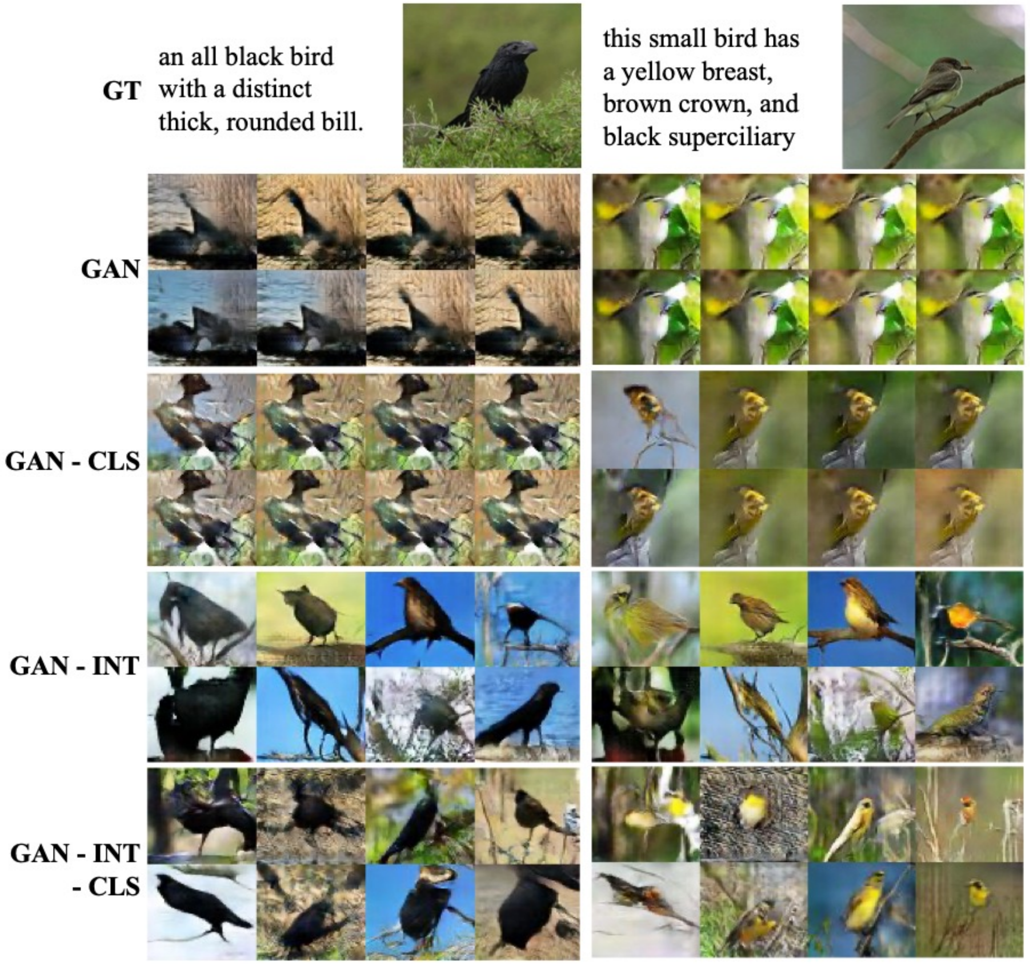
"Deep Visual-Semantic Alignments for Generating Image Descriptions" (Karpathy and Fei-Fei, CVPR 2015)

Translation (text to image)



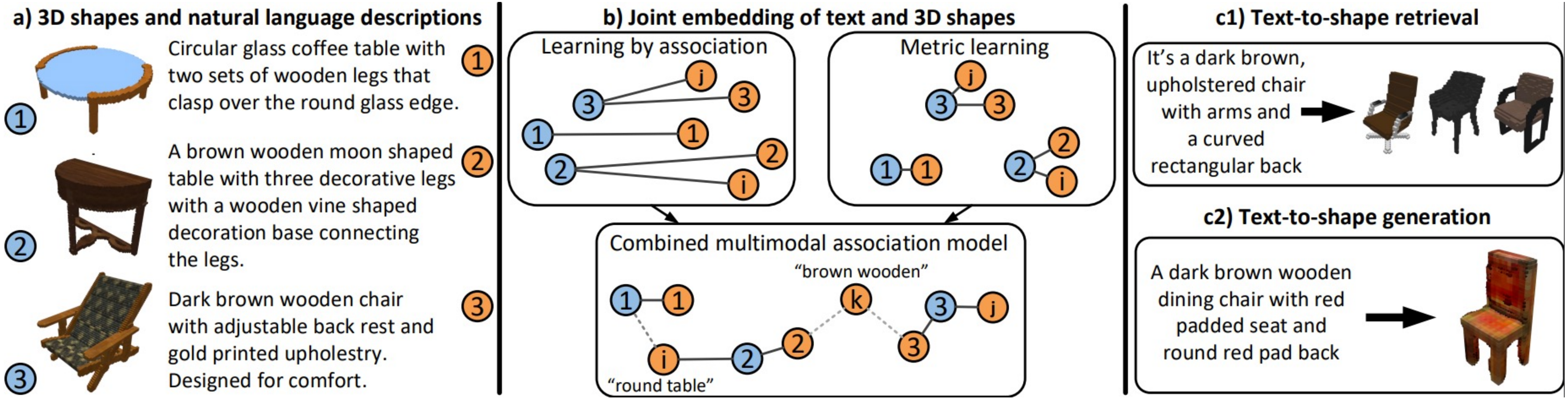
"Generative Adversarial Text to Image Synthesis" (Reed et al, ICML 2016)

Translation (text to image)



"Generative Adversarial Text to Image Synthesis" (Reed et al, ICML 2016)

Text and shape



Text2shape: Generating shapes from natural language by learning joint embeddings
Chen et al, ACCV 2018

Next time

- Paper presentations and discussion (Monday 1/24)
    - (Shichong) ViCo
    - (Han-Hung) CLIP
- Paper critiques due by midnight Sunday 1/23