Deep Metric Learning to Rank

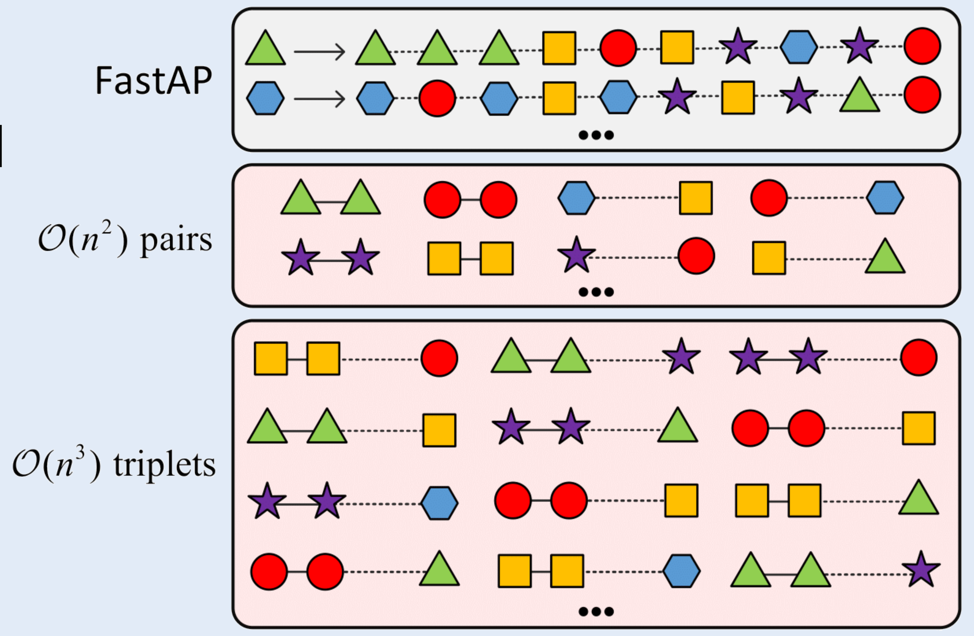
Fatih Cakir[1]* Kun He[2]* Xide Xia[3] Brian Kulis[3] Stan Sclaroff[3]
[1] FirstFuel [2] Facebook Reality Labs [3] Boston University (*equal contribution)

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

FastAP : Average Precision Loss

Motivation: Optimize the true "quantity of interest" for retrieval

✓ *list-wise* learning to rank



Reduce loss mis-specification

- Pair-based: *point-wise* LTR
- Triplet-based: *pair-wise* LTR

❖ Deep Euclidean embeddings, optimized wrt. Average Precision

Challenges

- Discrete sorting: gradients are zero almost everywhere
- AP: non-decomposable over individual examples

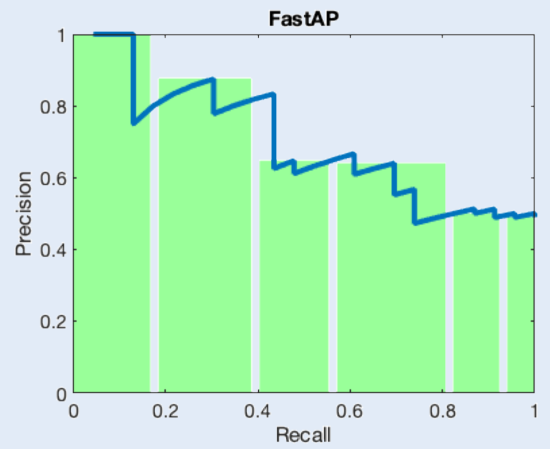
Contributions

✓ Approximation by quantization: well-behaved gradients
✓ Stochastic (minibatch) backprop
✓ Minibatch sampling strategy

FastAP : Formulation

Probabilistic interpretation of AP (AUC-PR)

- Parametric forms of precision and recall
- Change-of-variable + distance quantization
- Simple histogram-based formula



$$\mathrm{AP} = \int_\Omega \overbrace{P(\mathcal{R}^+|\mathcal{Z} < z)}^{\text{Precision}}\, d\, \overbrace{P(\mathcal{Z} < z|\mathcal{R}^+)}^{\text{Recall}}$$

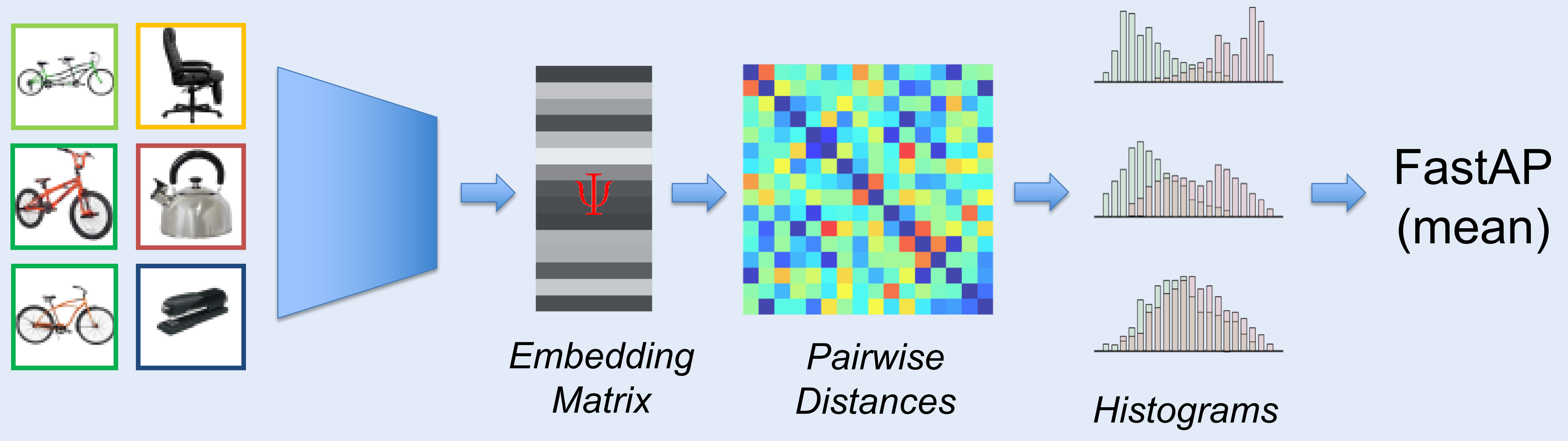
$$= \int_\Omega \frac{F(z|\mathcal{R}^+)P(\mathcal{R}^+)}{F(z)} p(z|\mathcal{R}^+)\, dz = \cdots \approx \frac{1}{N^+} \sum_{z=z_1,\ldots,z_L} \frac{H_z^+ h_z^+}{H_z}$$

Related work (partial list)

Metric Learning to Rank. B. McFee & G. Lanckriet, ICML'10
Hashing as Tie-Aware Learning to Rank. K. He et al., CVPR'18
Efficient Optimization of Rank-based Loss Functions. P. Mohapatra et al., CVPR'18

How to Train with Minibatches?

- Batch IR setup: fixed query set, fixed database - infeasible
- Minibatch setup:
  - Each example is treated as the query once.
  - Optimize mAP over minibatch.



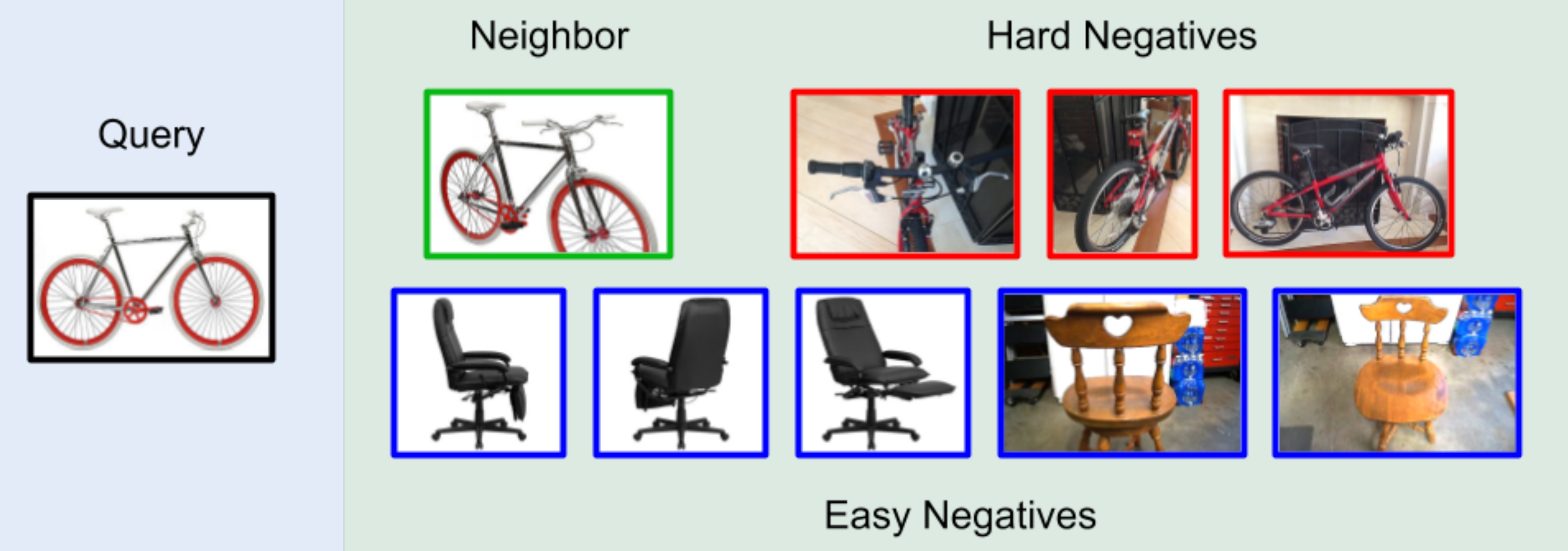
$$\frac{\partial \mathrm{FastAP}}{\partial \Psi_B} = -\frac{2}{M} \Psi_B \sum_{l=1}^{L} \left(F_l^+ B_l^+ + B_l^+ F_l^+ + F_l^- B_l^- + B_l^- F_l^-\right)_{M\times M}$$

F_l^+, F_l^- : diagonal. Time complexity : $\mathcal{O}(LM^2)$.

Does Sampling Matter?

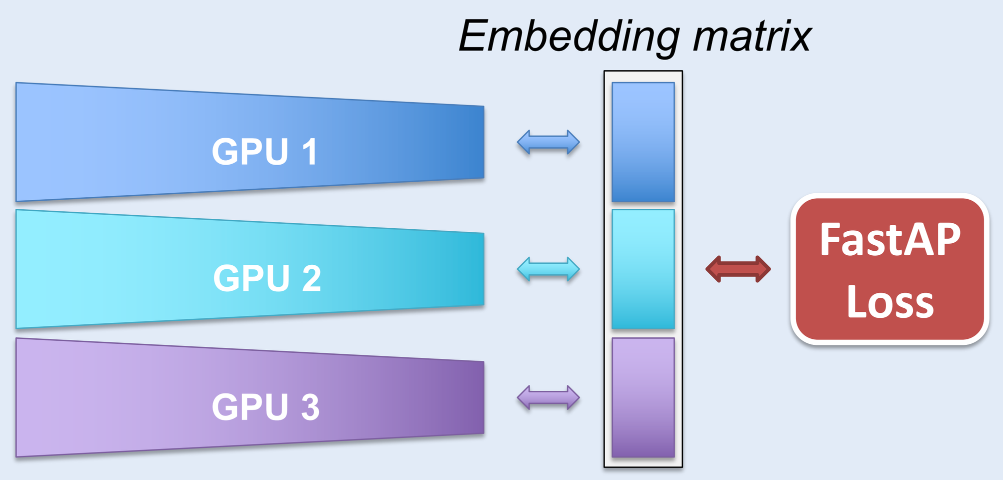
Yes. Need to construct "hard" retrieval problems in minibatches.

- We use side information: category (meta-class) labels
  - Classes in the same category are more similar!
- *Future work: automatic hard batch mining*

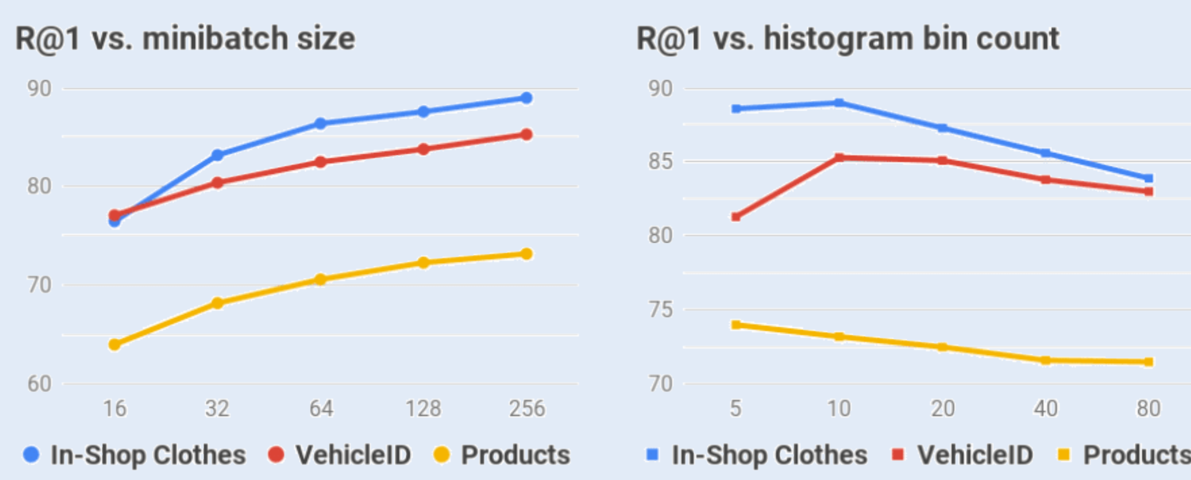


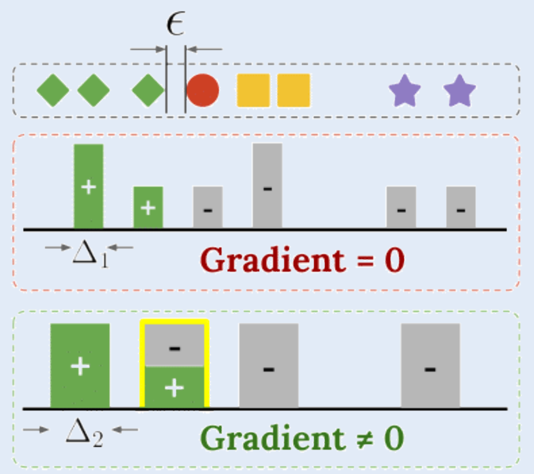
What about Batch Size?

- Larger batches → longer lists → harder retrieval problems
- Overcoming GPU mem. limit
  - Gather gradients wrt. embedding matrix
  - Also works on single GPU!



How Well Does It Work?





† Ensemble Method

Stanford Online Products		Dim.	R@1	R@10	R@100	R@1000		
ICCV'17	Margin	128	72.7	86.2	93.8	98.0		
arXiv'18	A-BIER †	512	74.2	86.9	94.8	98.2		
ECCV'18	Hierarchical Triplets	512	74.8	88.3	94.8	98.4		
ECCV'18	ABE-8 †	512	76.3	88.4	94.8	98.2		
CVPR'19	Divide and Conquer	128	75.9	88.4	94.9	98.1		
CVPR'19	Ranked List Loss	512	76.1	**89.1**	**95.4**	–		
	(Multi-level Ensemble)†	*512×3*	*79.8*	*91.3*	*96.3*	–		
FastAP	ResNet-50, $M = 96$	512	75.8	**89.1**	**95.4**	**98.5**		
FastAP	ResNet-50, $M = 256$*	512	**76.4**	89.0	95.1	98.2		
In-Shop Clothes Retrieval		Dim.	R@1	R@10	R@20	R@30	R@40	R@50
ECCV'18	DREML †	192×48	78.4	93.7	95.8	96.7	–	–
ECCV'18	Hierarchical Triplets	128	80.9	94.3	95.8	97.2	97.4	97.8
arXiv'18	A-BIER †	512	83.1	95.1	96.9	97.5	97.8	98.0
ECCV'18	ABE-8 †	512	87.3	96.7	97.9	98.2	98.5	98.7
CVPR'19	Divide and Conquer	128	85.7	95.5	96.9	97.5	–	98.0
FastAP	ResNet-18, $M = 256$	512	89.0	97.2	98.1	98.5	98.7	98.9
FastAP	ResNet-50, $M = 256$*	512	**90.9**	**97.7**	**98.5**	**98.8**	**98.9**	**99.1**
PKU VehicleID		Dim.	R@1(S)	R@5(S)	R@1(M)	R@5(M)	R@1(L)	R@5(L)
CVPR'16	Mixed Diff+CCL	1024	49.0	73.5	42.8	66.8	38.2	61.6
arXiv'18	A-BIER †	512	86.3	92.7	83.3	88.7	81.9	88.7
ECCV'18	DREML †	192×12	88.5	94.8	87.2	94.2	83.1	92.4
CVPR'19	Divide and Conquer	128	87.7	92.9	85.7	90.4	82.9	90.2
FastAP	ResNet-18, $M = 256$	512	90.9	96.0	88.9	95.2	85.3	93.9
FastAP	ResNet-50, $M = 256$*	512	**91.9**	**96.8**	**90.6**	**95.9**	**87.5**	**95.1**