Generative Adversarial Networks for Information Retrieval

Presented by Lantao Yu

April 19, 2017

SIGIR 2017

- **IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models**
  Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang and Dell Zhang

- **Deep Semantic Hashing with Generative Adversarial Networks**
  Zhaofan Qiu, Yingwei Pan, Ting Yao and Tao Mei

SIGIR 2017 TOKYO • JAPAN

IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models

Jun Wang
University College London
j.wang@cs.ucl.ac.uk

Lantao Yu, Weinan Zhang
Shanghai Jiao Tong University
wnzhang@sjtu.edu.cn

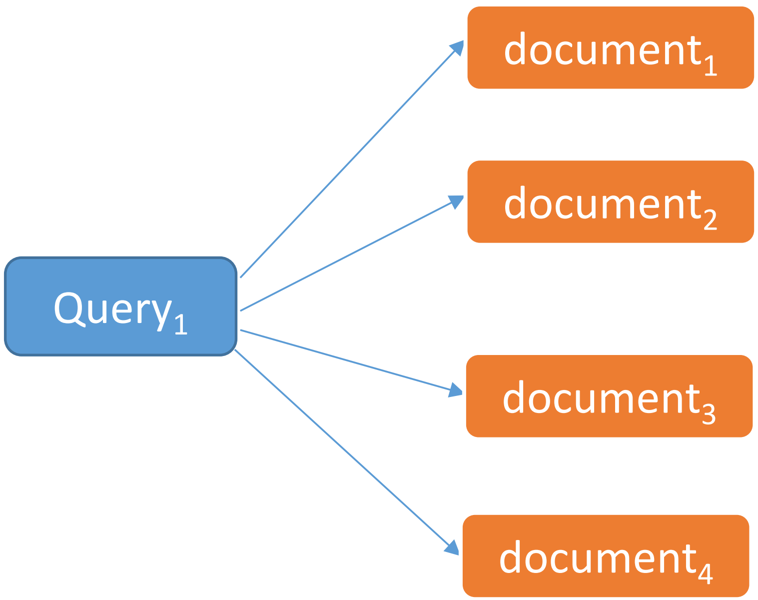
Yu Gong, Yinghui Xu
Alibaba Inc.
renji.xyh@taobao.com

Benyou Wang, Peng Zhang
Tianjin University
pzhang@tju.edu.cn

Dell Zhang
Birkbeck, University of London
dell.z@ieee.org

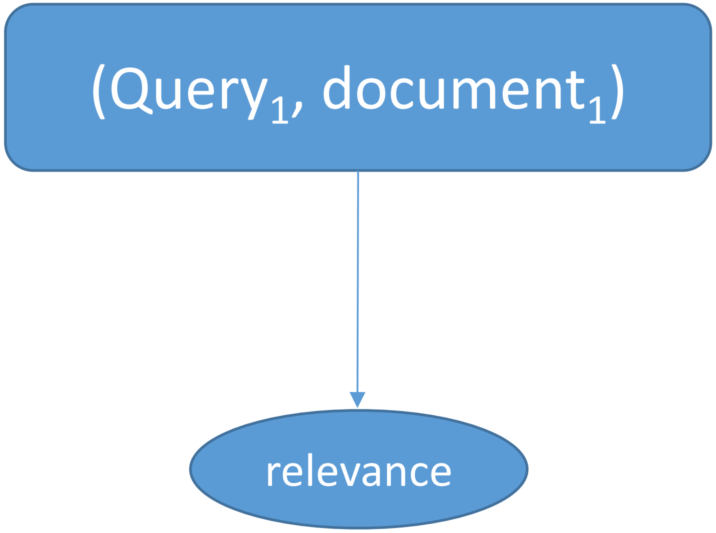
Two schools of thinking in IR modeling

Generative Retrieval



- Assume there is an underlying stochastic generative process between documents and queries
- Generate/Select relevant documents given a query

Discriminative Retrieval



- Learns from labeled relevant judgments
- Predict the relevance given a query-document pair

Three paradigms in Learning to Rank (LTR)

- **Pointwise:** learn to approximate the relevance estimation of each document to the human rating
- **Pairwise:** distinguish the more-relevant document from a document pair
- **Listwise:** learn to optimise the (smoothed) loss function defined over the whole ranking list for each query

IRGAN: A minimax game unifying both models

- Take advantage of both schools of thinking:
  - The generative model learns to fit the relevance distribution over documents via the signal from the discriminative model.
  - The discriminative model is able to exploit the unlabeled data selected by the generative model to achieve a better estimation for document ranking.

IRGAN Formulation

- The underlying true relevance distribution $p_{\text{true}}(d|q,r)$ depicts the user's relevance preference distribution over the candidate documents with respect to his submitted query
- Training set: A set of samples from $p_{\text{true}}(d|q,r)$
- Generative retrieval model $p_\theta(d|q,r)$
  - Goal: approximate the true relevance distribution
- Discriminative retrieval model $f_\phi(q,d)$
  - Goal: distinguish between relevant documents and non-relevant documents

IRGAN Formulation

- Overall Objective

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\mathrm{true}}(d|q_n,r)} \left[\log D(d|q_n)\right] + \mathbb{E}_{d \sim p_{\theta}(d|q_n,r)} \left[\log(1 - D(d|q_n))\right] \Big)$$

where $D(d|q) = \sigma(f_{\phi}(d,q)) = \dfrac{\exp(f_{\phi}(d,q))}{1+\exp(f_{\phi}(d,q))}$

IRGAN Formulation

- Optimizing Discriminative Retrieval

$$\phi^* = \arg\max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n, r)} \left[\log(\sigma(f_\phi(d, q_n))) \right] +$$

$$\mathbb{E}_{d \sim p_{\theta^*}(d|q_n, r)} \left[\log(1 - \sigma(f_\phi(d, q_n))) \right] \Big)$$

IRGAN Formulation

- Optimizing Generative Retrieval
  - Samples documents from the whole document set to fool its opponent

$$\theta^* = \arg\min_{\theta} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n,r)} \left[\log \sigma(f_\phi(d,q_n))\right] + \mathbb{E}_{d \sim p_\theta(d|q_n,r)} \left[\log(1 - \sigma(f_\phi(d,q_n)))\right] \Big)$$

$$= \arg\max_{\theta} \sum_{n=1}^{N} \underbrace{\mathbb{E}_{d \sim p_\theta(d|q_n,r)} \boxed{\left[\log(1 + \exp(f_\phi(d,q_n)))\right]}}_{\text{denoted as } J^G(q_n)} \quad \text{Reward Term}$$

- REINFORCE (Advantage Function)

IRGAN Formulation

- Algorithm

Algorithm 1 Minimax Game for IR (a.k.a IRGAN)

Input: generator $p_\theta(d|q, r)$; discriminator $f_\phi(\boldsymbol{x}_i^q)$;
training dataset $\mathcal{S} = \{\boldsymbol{x}\}$

Initialise $p_\theta(d|q, r)$, $f_\phi(q, d)$ with random weights θ, ϕ.
Pre-train $p_\theta(d|q, r)$, $f_\phi(q, d)$ using $\mathcal{S}$
repeat
  for g-steps **do**
    $p_\theta(d|q, r)$ generates K documents for each query q
    Update generator parameters via policy gradient Eq. (5)
  end for
  for d-steps **do**
    Use current $p_\theta(d|q, r)$ to generate negative examples and combine with given positive examples $\mathcal{S}$
    Train discriminator $f_\phi(q, d)$ by Eq. (3)
  end for
until IRGAN converges

IRGAN Formulation

- Extension to Pairwise Case
  - It is common that the dataset is a set of ordered document pairs for each query rather than a set of relevant documents.
  - Capture relative preference judgements rather than absolute relevance judgements

- Now, for each query q_n ,we have a set of labelled document pairs $R_n = \{\langle d_i, d_j \rangle | d_i \succ d_j\}$

IRGAN Formulation

- Extension to Pairwise Case
  - Discriminator would try to predict if a document pair is correctly ranked, which R_n can be implemented as many pairwise ranking loss function:
    - RankNet: $\log(1 + \exp(-z))$
    - Ranking SVM (Hinge Loss): $(1 - z)_+$
    - RankBoost: $\exp(-z)$

  where $z = f_\phi(d_u, q) - f_\phi(d_v, q)$

IRGAN Formulation

- Extension to Pairwise Case
  - Generator would try to generate document pairs that are similar to those in R_n , i.e., with the correct ranking.
  - A softmax function over the Cartesian Product of the document sets, where the logits is the advantage of d_i over d_j in a document pair (d_i, d_j)

An Intuitive Explanation of IRGAN

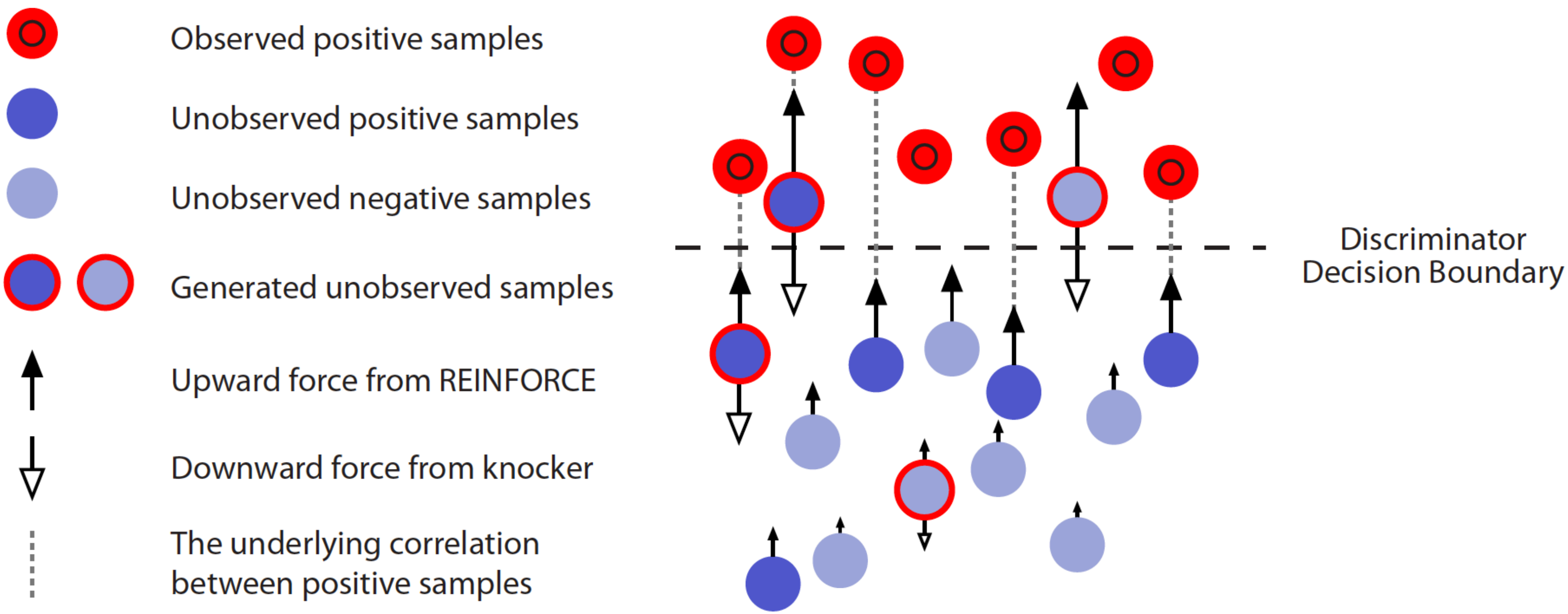


Figure 1: An illustration of IRGAN training.

An Intuitive Explanation of IRGAN

- The generative retrieval model is guided by the signal provided from the discriminative retrieval model, which makes it more favorable than the non-learning methods or the Maximum Likelihood Estimation (MLE) scheme.

- The discriminative retrieval model could be enhanced to better rank top documents via a strategic negative sampling from the generator.

Experiments: Web Search

Table 1: Webpage ranking performance comparison on MQ2008-semi dataset, where $*$ means significant improvement in a Wilcoxon signed-rank test.

	P@3	P@5	P@10	MAP
MLE	0.1556	0.1295	0.1029	0.1604
RankNet [3]	0.1619	0.1219	0.1010	0.1517
LambdaRank [5]	0.1651	0.1352	0.1076	0.1658
LambdaMART [4]	0.1368	0.1026	0.0846	0.1288
IRGAN-pointwise	0.1714	0.1657	**0.1257**	**0.1915**
IRGAN-pairwise	**0.2000**	**0.1676**	0.1248	0.1816
Impv-pointwise	3.82%	22.56%*	16.82%*	15.50%*
Impv-pairwise	21.14%*	23.96%*	15.98%	9.53%
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.1893	0.1854	0.2054	0.3194
RankNet [3]	0.1801	0.1709	0.1943	0.3062
LambdaRank [5]	0.1926	0.1920	0.2093	0.3242
LambdaMART [4]	0.1573	0.1456	0.1627	0.2696
IRGAN-pointwise	0.2065	**0.2225**	**0.2483**	**0.3508**
IRGAN-pairwise	**0.2148**	0.2154	0.2380	0.3322
Impv-pointwise	7.22%	15.89%	18.63%	8.20%
Impv-pairwise	11.53%	12.19%	13.71%	2.47%

Experiments: Web Search

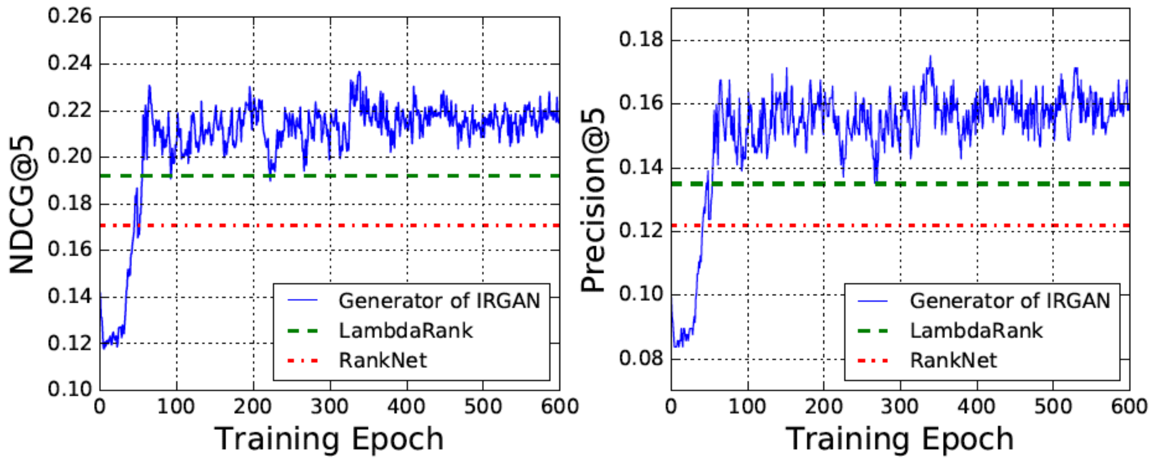


Figure 2: Learning curves of the pointwise IRGAN on web search task.

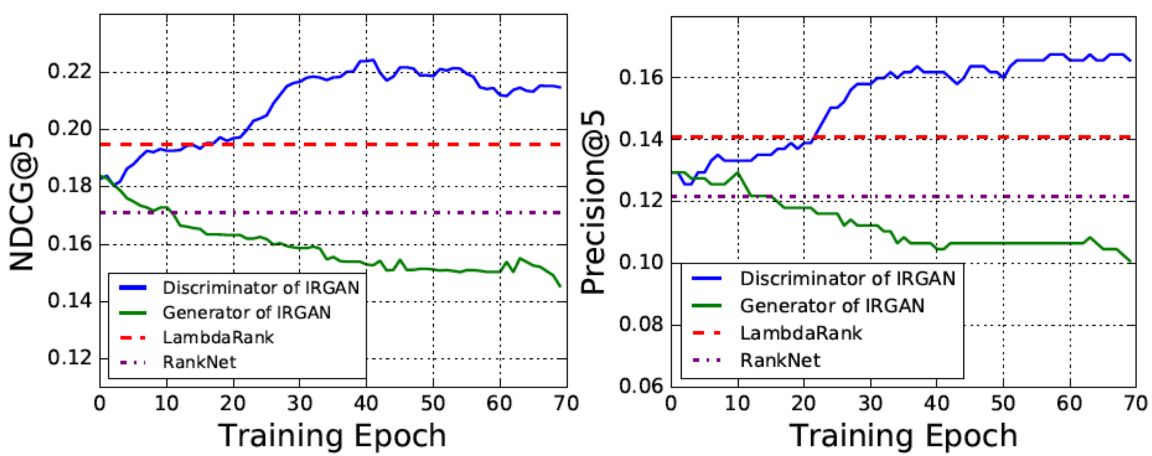


Figure 3: Learning curves of the pairwise IRGAN on web search task.

Experiments: Item Recommendation

Table 3: Item recommendation results (Movielens).

	P@3	**P@5**	**P@10**	**MAP**
MLE	0.3369	0.3013	0.2559	0.2005
BPR [35]	0.3289	0.3044	0.2656	0.2009
LambdaFM [45]	0.3845	0.3474	0.2967	0.2222
IRGAN-pointwise	0.4072	0.3750	0.3140	0.2418
Impv-pointwise	5.90%*	7.94%*	5.83%*	8.82%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3461	0.3236	0.3017	0.5264
BPR [35]	0.3410	0.3245	0.3076	0.5290
LambdaFM [45]	0.3986	0.3749	0.3518	0.5797
IRGAN-pointwise	0.4222	0.4009	0.3723	0.6082
Impv-pointwise	5.92%*	6.94%*	5.83%*	4.92%*

Table 4: Item recommendation results (Netflix).

	P@3	**P@5**	**P@10**	**MAP**
MLE	0.2941	0.2945	0.2777	0.0957
BPR [35]	0.3040	0.2933	0.2774	0.0935
LambdaFM [45]	0.3901	0.3790	0.3489	0.1672
IRGAN-pointwise	0.4456	0.4335	0.3923	0.1720
Impv-pointwise	14.23%*	14.38%*	12.44%*	2.87%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3032	0.3011	0.2878	0.5085
BPR [35]	0.3077	0.2993	0.2866	0.5040
LambdaFM [45]	0.3942	0.3854	0.3624	0.5857
IRGAN-pointwise	0.4498	0.4404	0.4097	0.6371
Impv-pointwise	14.10%*	14.27%*	13.05%*	8.78%*

Experiments: Item Recommendation

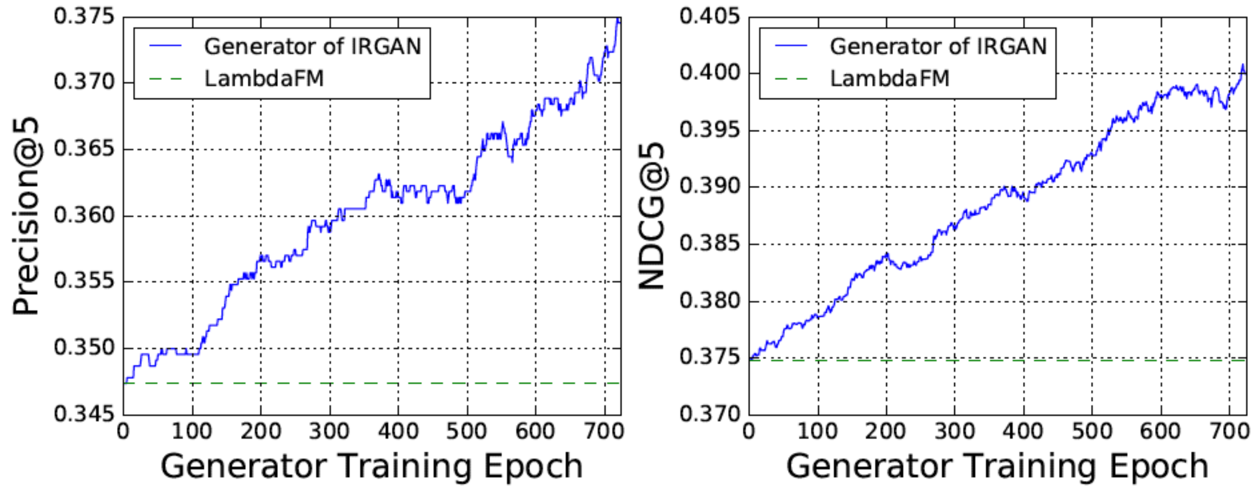


Figure 6: Learning curve of precision and NDCG of the generative retrieval model for top-5 item recommendation task on Movielens dataset.

Experiments: Question Answering

Table 5: The Precision@1 of InsuranceQA.

	test-1	test-2
QA-CNN [9]	0.6133	0.5689
LambdaCNN [9, 49]	0.6183	0.5838
IRGAN-pairwise	0.6383	0.5978
Impv-pairwise	3.23%*	2.74%

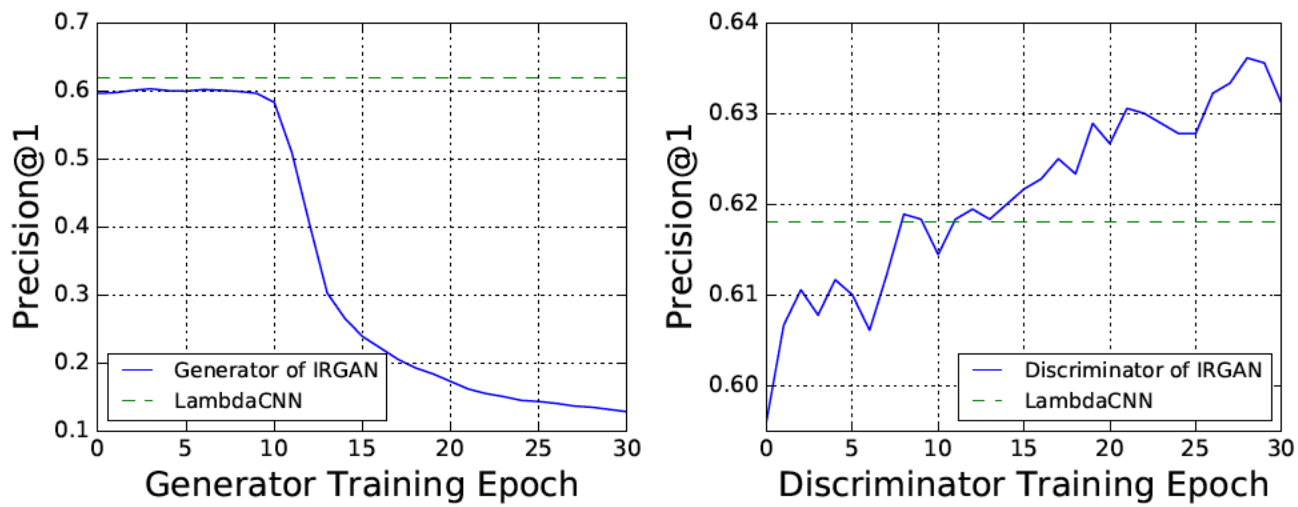


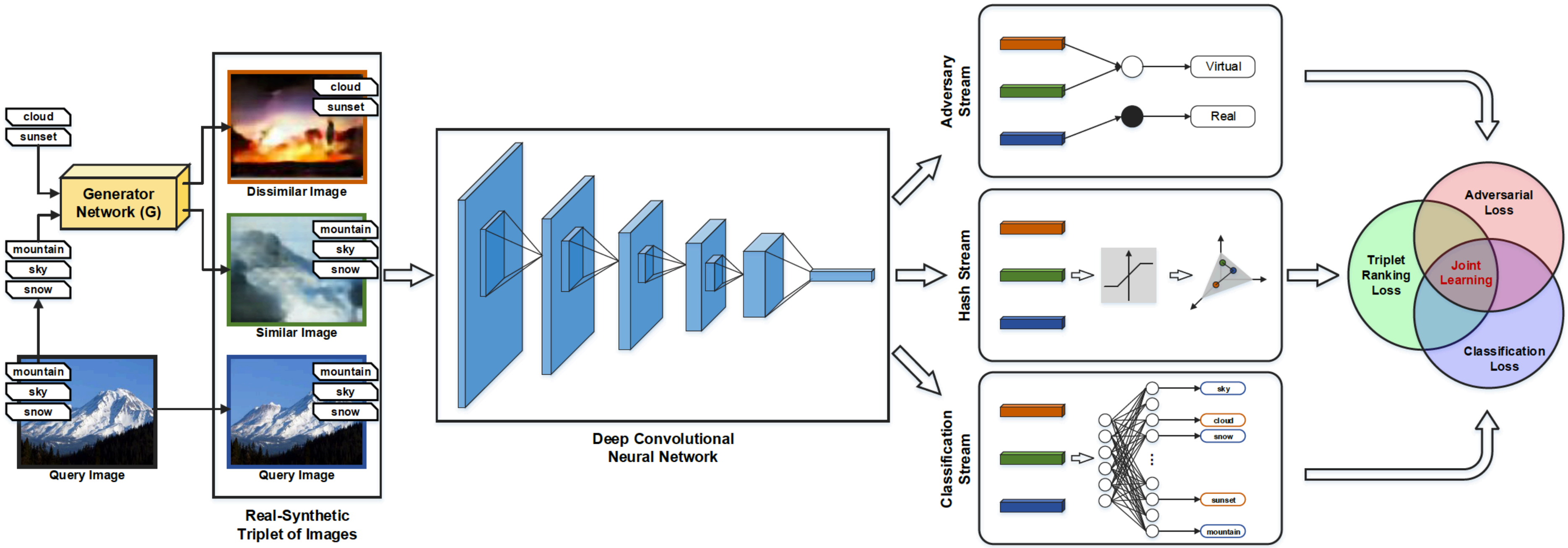
Figure 8: The experimental results in QA task.

Summary

- We proposed IRGAN framework that unifies two schools of information retrieval methodologies, via adversarial training in a minimax game, which takes advantage of both schools of thinking.
- Significant performance gains were observed in three typical information retrieval tasks.
- Experiments suggest that different equilibria could be reached in the end depending on the tasks and settings.

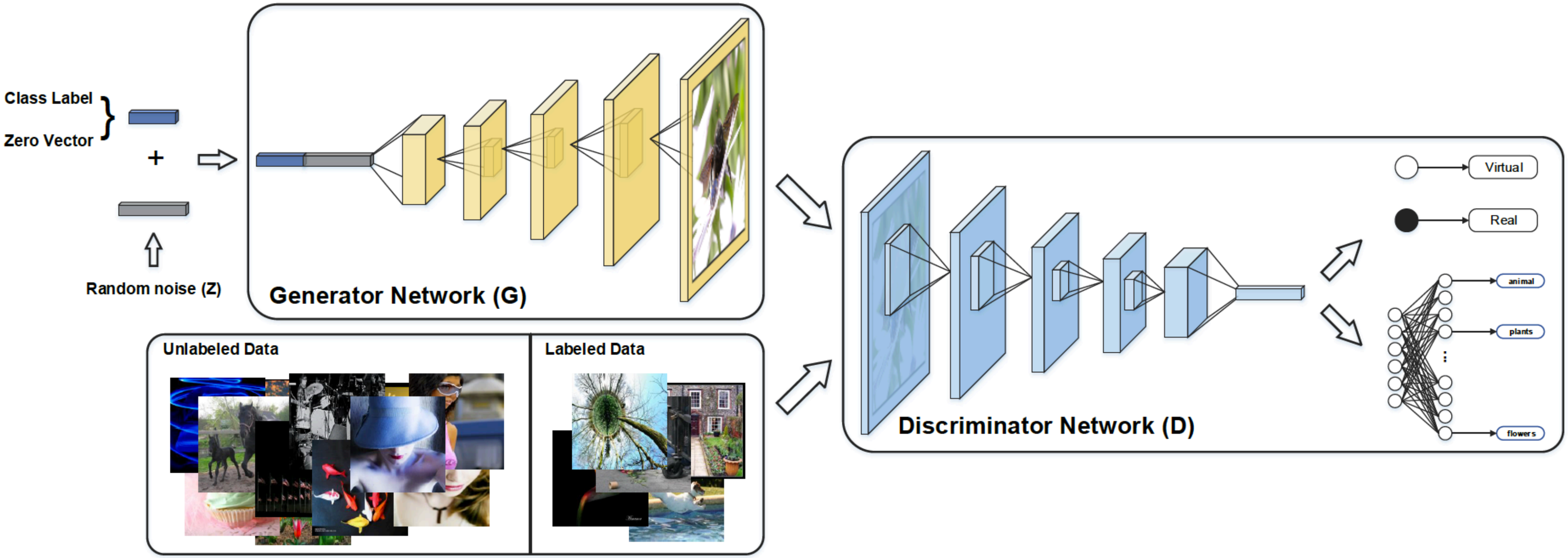
Deep Semantic Hashing with Generative Adversarial Networks

Zhaofan Qiu, Yingwei Pan, Ting Yao and Tao Mei



- A shared CNN for learning image representations
- An adversary stream for distinguishing synthetic images from real ones
- A hash stream for encoding each image into hash codes
- A classification stream for leveraging semantic supervision

Semi-supervised GANs



A semi-supervised GANs is first devised to leverage both unlabeled and labeled images for producing synthetic images conditioning on class labels.

$$J^D = l_c(x) + l_a(x)$$
$$J^G = l_c(x) - l_a(x)$$

where $l_a(x) = \begin{cases} -\log P(S = real|x), x \in \mathcal{X} \\ -\log P(S = synthetic|x), x \in \mathcal{X}_{syn} \end{cases}$

Hash Stream

- Hash stream is trained with the input real-synthetic triplets in a triplet-wise manner

$$
\begin{aligned}
&\hat{l}_{triplet}(x, x^{+}_{syn}, x^{-}_{syn}) \\
&= \max(0, 1 - \left\| \mathcal{H}(x) - \mathcal{H}(x^{-}_{syn}) \right\|_{H} + \left\| \mathcal{H}(x) - \mathcal{H}(x^{+}_{syn}) \right\|_{H}) \\
&s.t. \quad \mathcal{H}(x), \mathcal{H}(x^{+}_{syn}), \mathcal{H}(x^{-}_{syn}) \in \{0, 1\}^{K}
\end{aligned}
$$

Adversary Stream

- Adversary stream recognizes the label of synthetic or real for each image example.

$$\hat{l}_a(x, x_{syn}^{+}, x_{syn}^{-}) = \frac{1}{3}\left(l_a\left(x\right) + l_a\left(x_{syn}^{+}\right) + l_a\left(x_{syn}^{-}\right)\right)$$

Classification Stream

- Classification stream reinforces the hash learning to preserve semantic structures on both real and synthetic images.

$$
\begin{aligned}
l_c(x) = -\sum_{j=1}^{c} \big[& I_{(\mathbf{C}_j=1)} \log\left(P(\mathbf{C}_j = 1|\mathbf{x})\right) \\
& + (1 - I_{(\mathbf{C}_j=1)}) log\left(1 - P(\mathbf{C}_j = 1|\mathbf{x})\right) \big]
\end{aligned}
$$

$$
P(\mathbf{C}_j = 1|\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\delta}_j^{\top}\mathbf{x}}}
$$

Joint Optimization

- For shared CNN, the overall objective:

$$\hat{l}_{CNN} = \sum_{\mathcal{T}} \left[\hat{l}_{triplet}(x, x^{+}_{syn}, x^{-}_{syn}) \right.$$
$$\left. + \hat{l}_a(x, x^{+}_{syn}, x^{-}_{syn}) + \hat{l}_c(x, x^{+}_{syn}, x^{-}_{syn})\right]$$

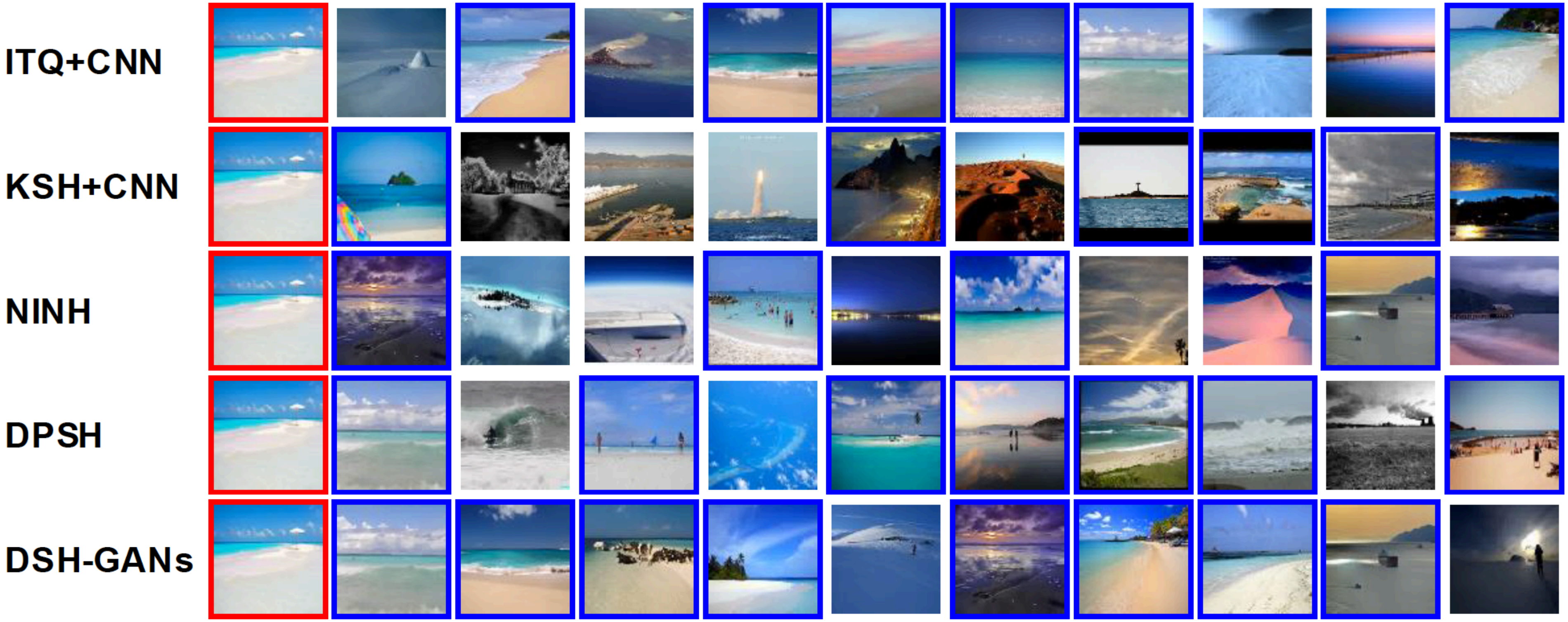
- For the generator network, the overall objective:

$$\hat{l}_{G} = \sum_{\mathcal{T}} \left[\hat{l}_{triplet}(x, x^{+}_{syn}, x^{-}_{syn}) \right.$$
$$\left. - \hat{l}_a(x, x^{+}_{syn}, x^{-}_{syn}) + \hat{l}_c(x, x^{+}_{syn}, x^{-}_{syn})\right]$$

Experiments

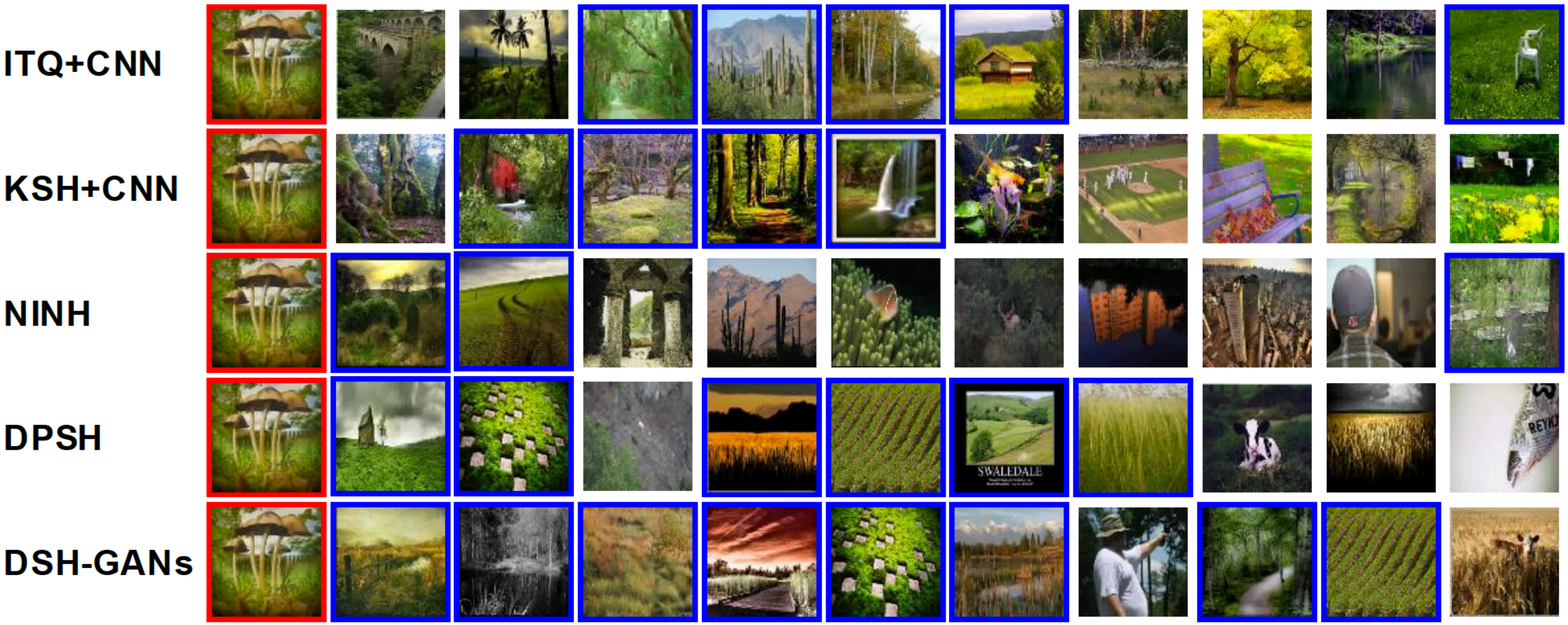
Method	CIFAR-10 (MAP)				NUS-WIDE (MAP)			
	12-bits	24-bits	32-bits	48-bits	12-bits	24-bits	32-bits	48-bits
DSH-GANs	**0.735**	**0.781**	**0.787**	**0.802**	**0.838**	**0.856**	**0.861**	**0.863**
DSH-GANs$^-$	0.726	0.769	0.772	0.783	0.823	0.847	0.845	0.854
DPSH	0.713	0.727	0.744	0.757	0.794	0.822	0.838	0.851
NINH	0.552	0.566	0.558	0.581	0.674	0.697	0.713	0.715
CNNH	0.439	0.476	0.472	0.489	0.611	0.618	0.625	0.608
KSH+CNN	0.446	0.502	0.518	0.516	0.746	0.774	0.765	0.749
ITQ+CNN	0.212	0.230	0.234	0.240	0.728	0.707	0.689	0.661
SH+CNN	0.158	0.157	0.154	0.151	0.620	0.611	0.620	0.591
LSH+CNN	0.134	0.157	0.173	0.185	0.438	0.586	0.571	0.507
KSH	0.303	0.337	0.346	0.356	0.556	0.572	0.581	0.588
ITQ	0.162	0.169	0.172	0.175	0.452	0.468	0.472	0.477
SH	0.127	0.128	0.126	0.129	0.454	0.406	0.405	0.400
LSH	0.121	0.126	0.120	0.120	0.403	0.421	0.426	0.441

Top 10 Image Retrieval Results by different methods in response to two query images



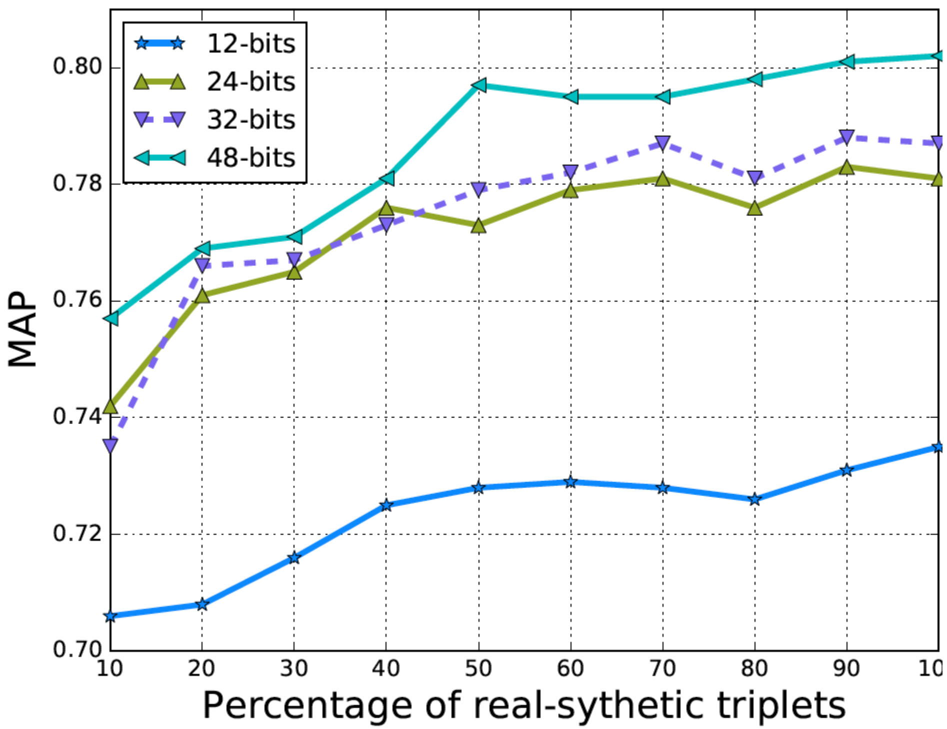
Blue Box: excellent ones whose annotations completely contain all the labels of the query images

Top 10 Image Retrieval Results by different methods in response to two query images

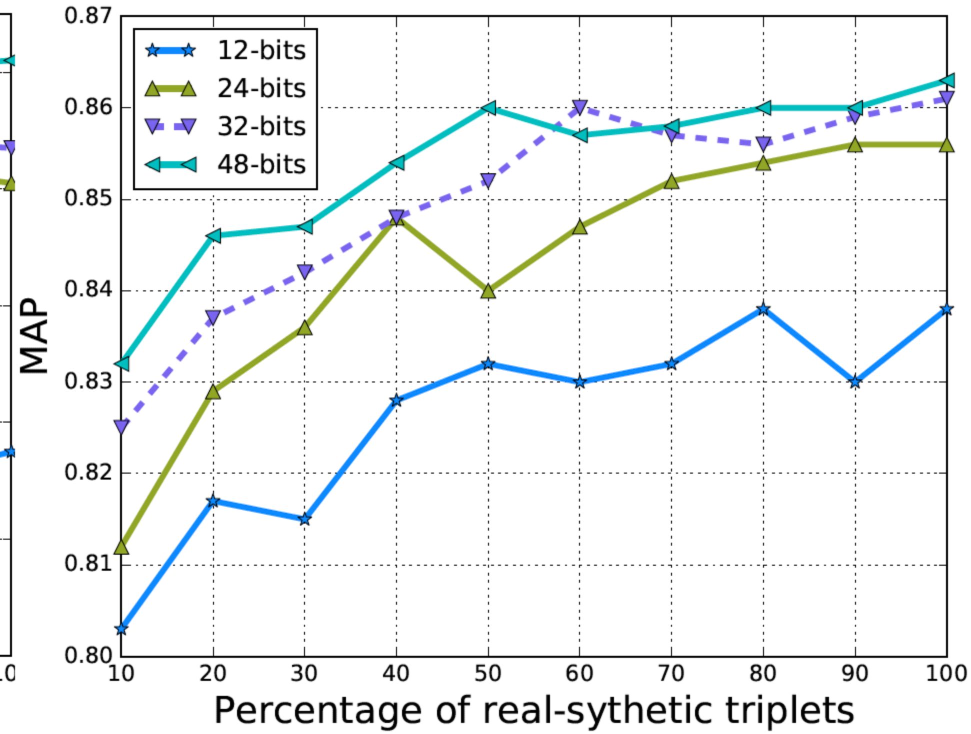


Blue Box: excellent ones whose annotations completely contain all the labels of the query images

In hashing stream, the similar and dissimilar image can be synthetic or real images. The ratio of synthetic images affects the performance:



(a) CIFAR-10.



(b) NUS-WIDE.

Visualization of synthetic images