Attention-aware deep adversarial hashing for cross-modal retrieval

Xi Zhang, Hanjiang Lai, and Jiashi Feng

ECCV 2018

Background

Cross-modal retrieval: takes one type of data as query, and returns the relevant data of another type (text, image, audio, video)

a) Real-valued representation learning

b) Binary representation learning

- ✓ Low storage cost and fast retrieval speed
- ✓ feature extraction -> common Hamming space
- ✓ unsupervised / pairwise-based / supervised

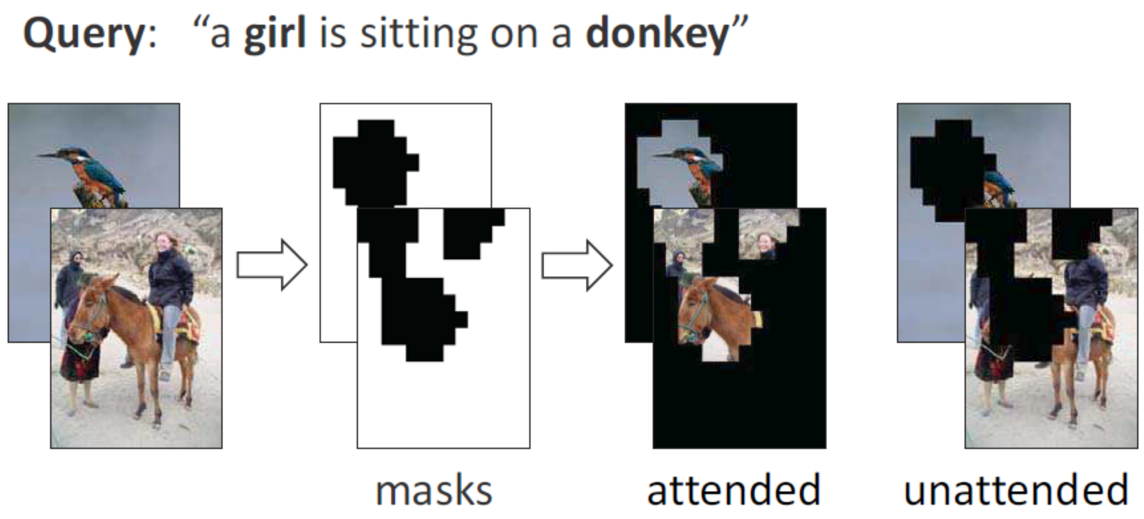
Problem Definition

- n training samples $\{I_i, T_i\}_{i=1}^{n}$
- I_i: the i-th image
- T_i: the corresponding text description of image I_i
- Cross-modal similarity matrix S
  - $S(i,j) = 1$, the i-th image and j-th text are similar
  - $S(i,j) = 0$, dissimilar

- Goal: learn two mapping functions to transform images and texts into a common binary codes space, in which similarities between the paired images and texts are preserved
  - $S(i,j) = 1$, the Hamming distance should be small.
  - $S(i,j) = 0$, the Hamming distance should be large

Attention-aware Deep Adversarial Hashing

- Idea: find the region of multi-modal data favoured for retrieval
- Attention-aware Deep Adversarial Hashing: enhance the measurement of content similarities by selectively focusing on the informative parts of multi-modal data



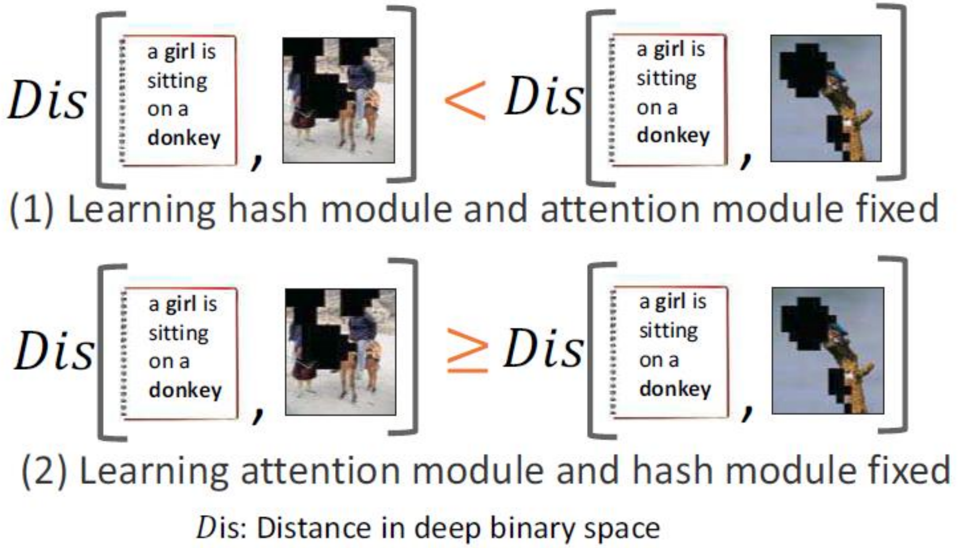
(I) The attention module

Attention-aware Deep Adversarial Hashing

- Three building blocks:
  - Feature learning module
  - Attention module
    - divide the feature representation into the attended and unattended feature representations.
  - Hashing module

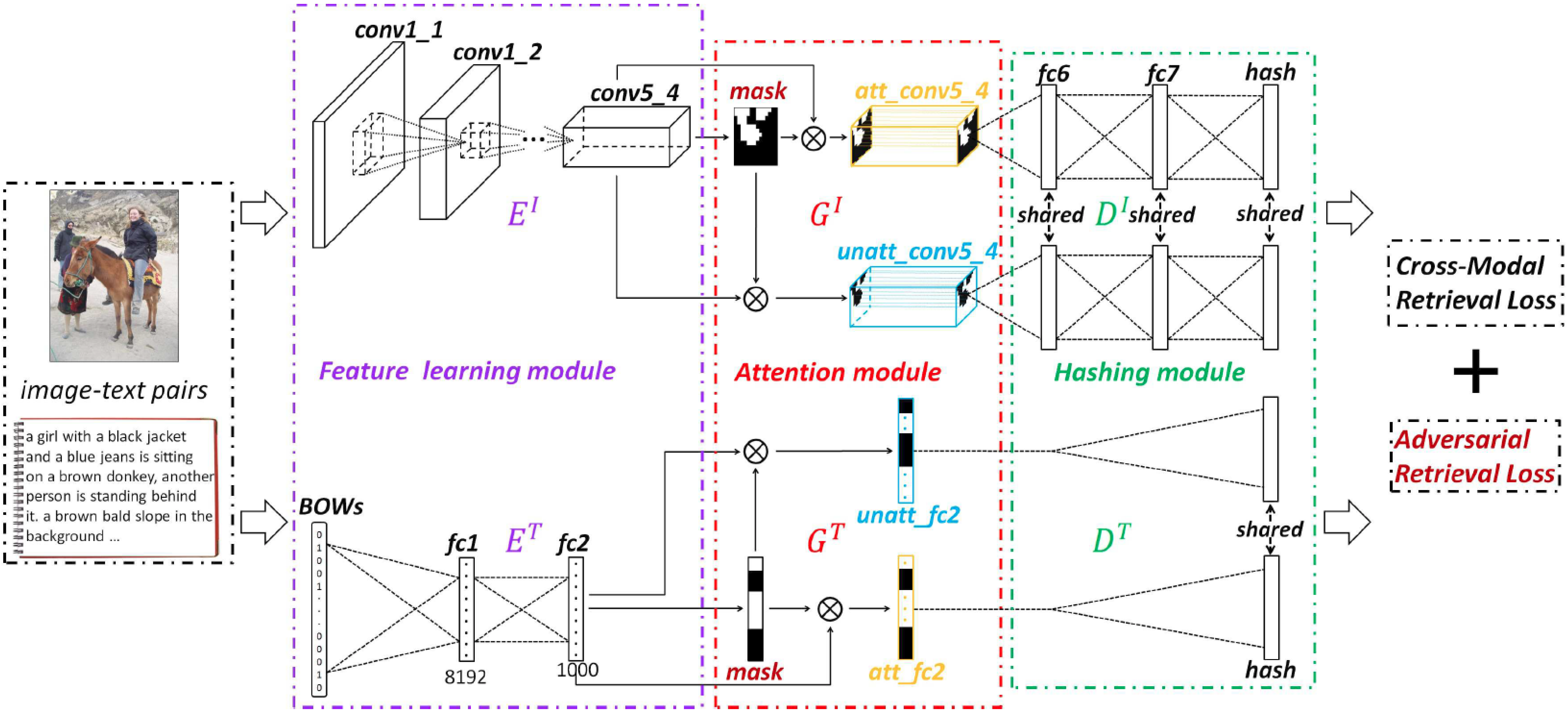
The attention module and hashing module are trained in an adversarial way:

1) The attention module attempts to make the hashing module unable to preserve the similarity of multi-modal data w.r.t. the unattended feature representations;
2) The hashing module aims to preserve the similarities of multi-modal data w.r.t. the attended and unattended feature representations.



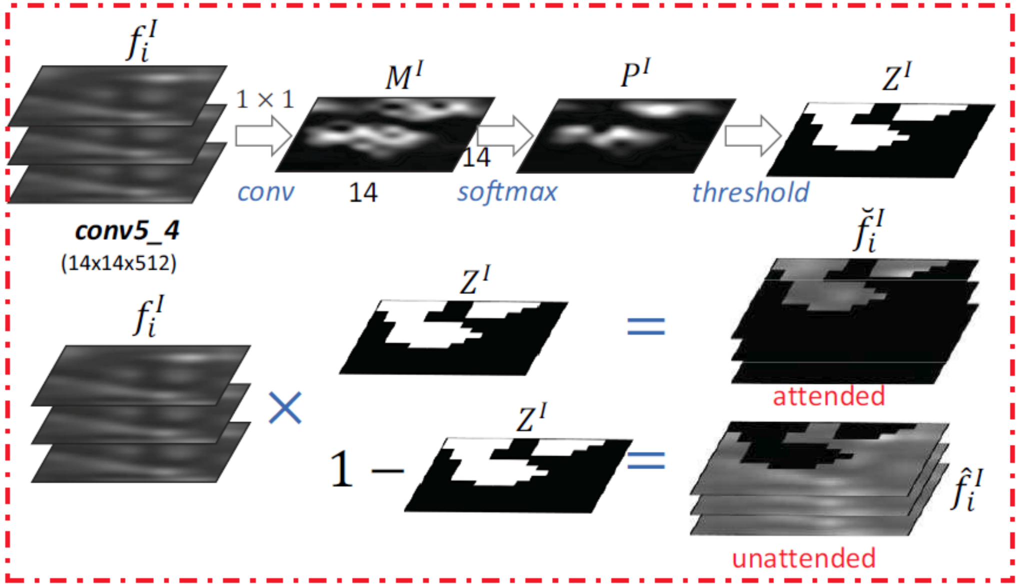
(II) Adversarial learning

Network Architecture

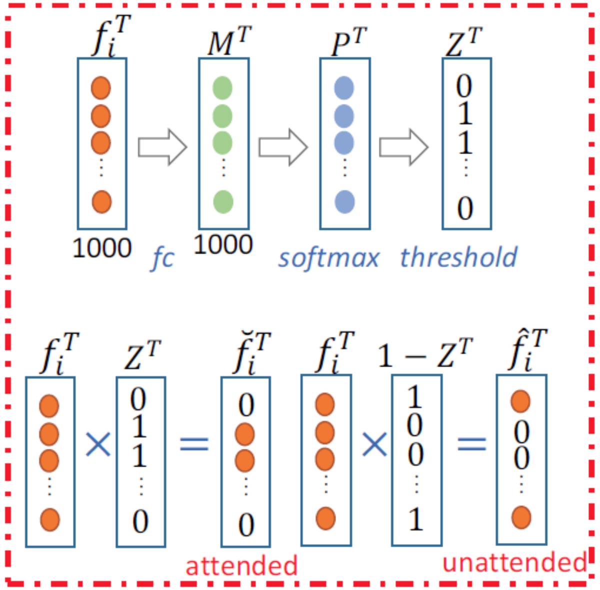


Network Architecture

- Feature learning module
  - E^I: VGGNet
  - E^T: Two-layer feed-forward neural network (BOW -> 8192 -> 1000)
- Attention module



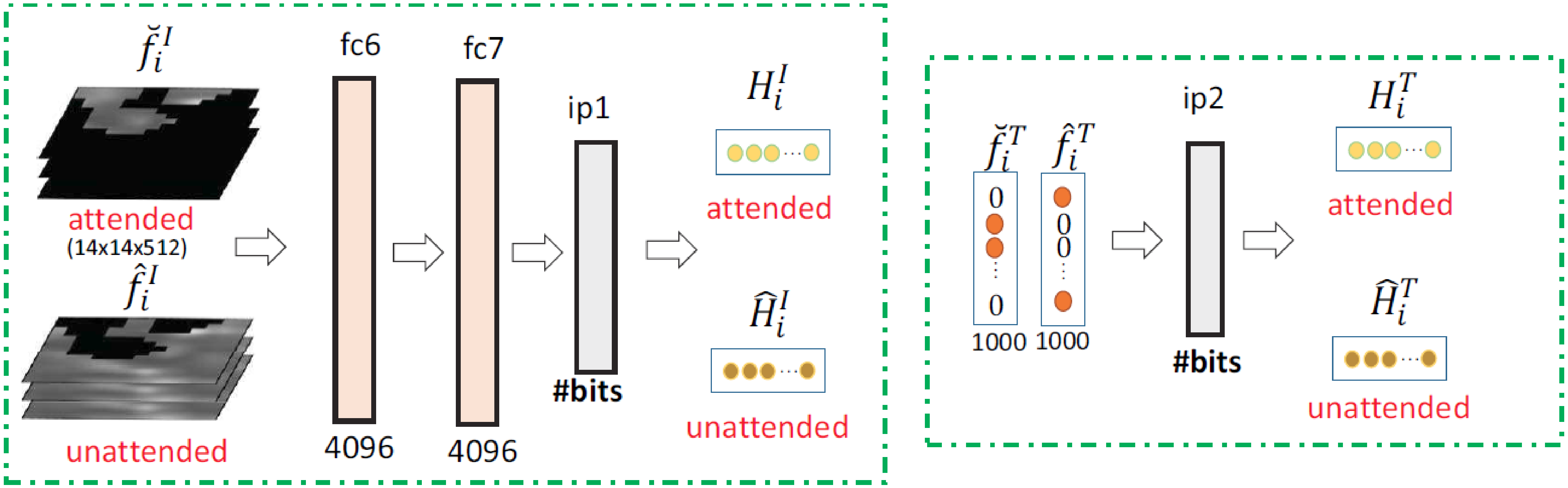
(I) The attention module for image: G^I



(II) The attention module for text: G^T

Network Architecture

- Hashing module



(I) The hashing module for image: D^I

(II) The hashing module for text: D^T

Objective function

① Cross-modal Retrieval Loss:

The inter-modal ranking loss + the intra ranking loss:

$$\min \mathcal{F}_{T\to I} + \mathcal{F}_{I\to T} + \mathcal{F}_{I\to I} + \mathcal{F}_{T\to T}$$

$$\mathcal{F}_{A\to B} = \sum_{\langle i,j,k\rangle} \max\{0, \varepsilon + ||H_i^A - H_j^B|| - ||H_i^A - H_k^B||\}$$

$$s.t. \qquad \forall\langle i,j,k\rangle,\ S(i,j) > S(i,k),$$

$$A \in \{T, I\}\ ,\ \ B \in \{T, I\}$$

② Adversarial Retrieval Loss:

Objective function

① Cross-modal Retrieval Loss:

② Adversarial Retrieval Loss:

$$\min_{D^I, D^T} \max_{G^I, G^T} \mathcal{F}_{T\to\hat{I}} + \mathcal{F}_{I\to\hat{T}}$$

$$\mathcal{F}_{T\to\hat{I}} + \mathcal{F}_{I\to\hat{T}} = \sum_{\langle i,j,k\rangle} \max\{0, \varepsilon + ||H_i^T - \hat{H}_j^I|| - ||H_i^T - \hat{H}_k^I||\}$$
$$+ \sum_{\langle i,j,k\rangle} \max\{0, \varepsilon + ||H_i^I - \hat{H}_j^T|| - ||H_i^I - \hat{H}_k^T||\}$$

Objective function

Full Objective:

$$\mathcal{F}(E^I, E^T, G^I, G^T, D^I, D^T) = \mathcal{F}_{T\rightarrow\hat{I}} + \mathcal{F}_{I\rightarrow\hat{T}} + \mathcal{F}_{T\rightarrow I} + \mathcal{F}_{I\rightarrow T} + \mathcal{F}_{I\rightarrow I} + \mathcal{F}_{T\rightarrow T}.$$

Train the model alternatively:

1. With the parameters in G^I and G^T fixed, train E^I, E^T, D^I, D^T (4 steps)

$$\min_{E^I, E^T, D^I, D^T} \mathcal{F}(E^I, E^T, G^I, G^T, D^I, D^T).$$

2. With the parameters in E^I, E^T, D^I, D^T fixed, train G^I, G^T (1 step)

$$\max_{G^I, G^T} \mathcal{F}_{T\rightarrow\hat{I}} + \mathcal{F}_{I\rightarrow\hat{T}}.$$

Experiments

- Datasets:
  - IAPR TC-12: 20,000 images, each image is associated with a text caption, 255 labels, 2912-d BOW vector
  - MIR-Flickr 25K: 25,000 multi-label images, each image is associated with several text tags (at least 20 textual tags), 1386-d BOW vector
  - NUS-WIDE: 269,648 images, each image is associated with one or multiple textural tags in 81 semantic concepts. -> 195,834 images belongs to 21 most frequent labels, 1000-d BOW vector
- Query sets: 2000 image-text pairs for IAPR TC-12/MIR-Flickr 25K, 2100 image-text pairs for NUS-WIDE
- Training sets: 10,000 pairs for IAPR TC-12/MIR-Flickr 25K, 10,500 pairs for NUS-WIDE

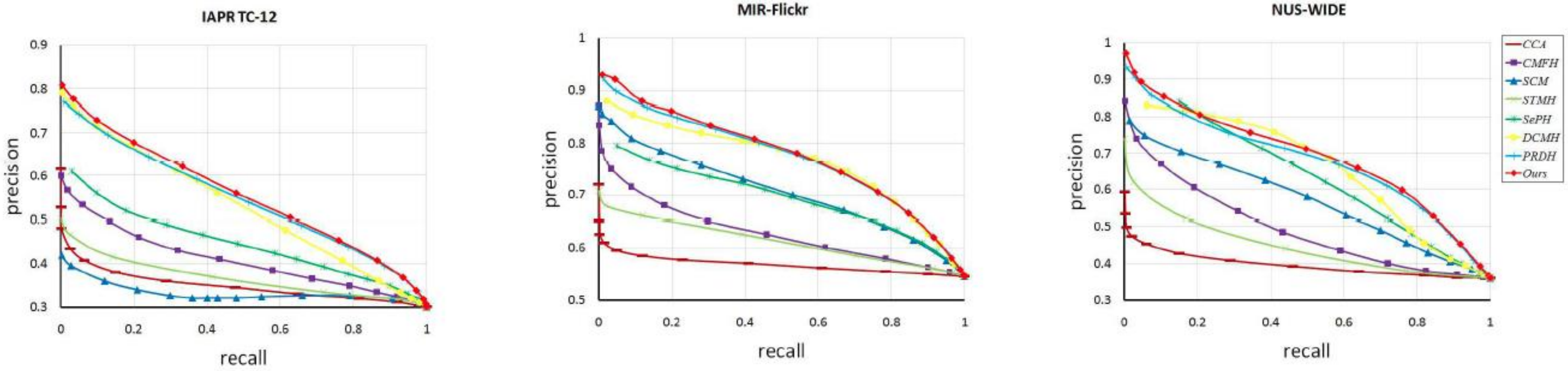
Comparison with state-of-the-art methods

mAP:

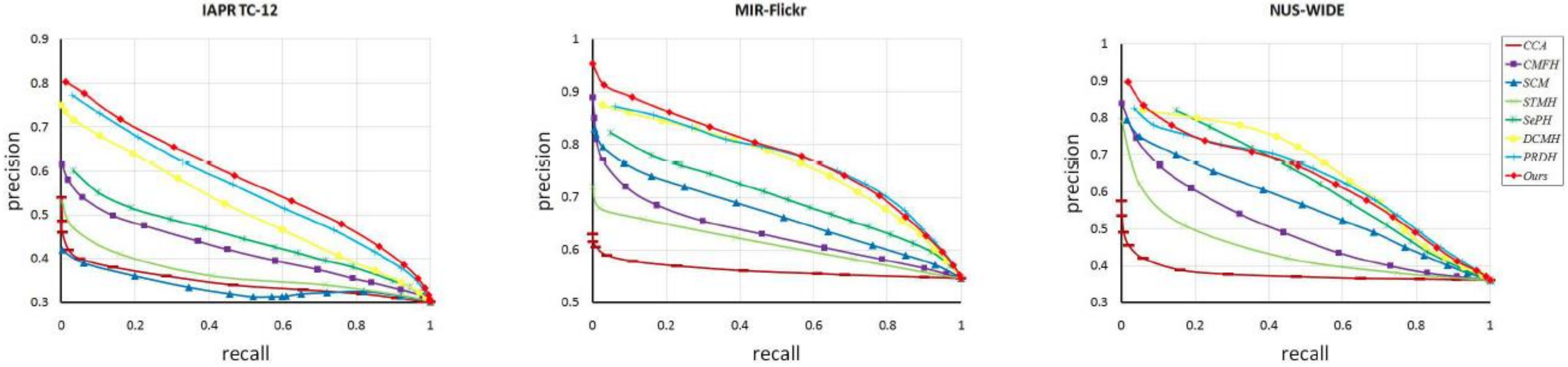
Task		IAPR TC-12			MIR-Flickr 25k			NUS-WIDE		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
Text ↓ Image	CCA	0.3493	0.3438	0.3378	0.5742	0.5713	0.5691	0.3731	0.3661	0.3613
	CMFH	0.4168	0.4212	0.4277	0.6365	0.6399	0.6429	0.5031	0.5187	0.5225
	SCM	0.3453	0.3410	0.3470	0.6939	0.7012	0.7060	0.5344	0.5412	0.5484
	STMH	0.3687	0.3897	0.4044	0.6074	0.6153	0.6217	0.4471	0.4677	0.4780
	SePH	0.4423	0.4562	0.4648	0.7216	0.7261	0.7319	0.5983	0.6025	0.6109
	DCMH	0.5185	0.5378	0.5468	0.7827	0.7900	0.7932	0.6389	0.6511	0.6571
	PRDH	0.5244	0.5434	0.5548	0.7890	0.7955	0.7964	0.6527	0.6916	0.6720
	Ours	**0.5358**	**0.5565**	**0.5648**	**0.7922**	**0.8062**	**0.8074**	**0.6789**	**0.6975**	**0.7039**
Image ↓ Text	CCA	0.3422	0.3361	0.3300	0.5719	0.5693	0.5672	0.3742	0.3667	0.3617
	CMFH	0.4189	0.4234	0.4251	0.6377	0.6418	0.6451	0.4900	0.5053	0.5097
	SCM	0.3692	0.3666	0.3802	0.6851	0.6921	0.7003	0.5409	0.5485	0.5553
	STMH	0.3775	0.4002	0.4130	0.6132	0.6219	0.6274	0.4710	0.4864	0.4942
	SePH	0.4442	0.4563	0.4639	0.7123	0.7194	0.7232	0.6037	0.6136	0.6211
	DCMH	0.4526	0.4732	0.4844	0.7410	0.7465	0.7485	0.5903	0.6031	0.6093
	PRDH	0.5003	0.4935	0.5135	0.7499	0.7546	0.7612	0.6107	**0.6302**	0.6276
	Ours	**0.5293**	**0.5283**	**0.5439**	**0.7563**	**0.7719**	**0.7720**	**0.6403**	0.6294	**0.6520**

Comparison with state-of-the-art methods

Precision-Recall Curves:



(a) Query from text to image task. ($T \rightarrow I$)



(b) Query from image to text task. ($I \rightarrow T$)

Comparison with state-of-the-art methods

Top-500 mAP

On IAPR TC-12:

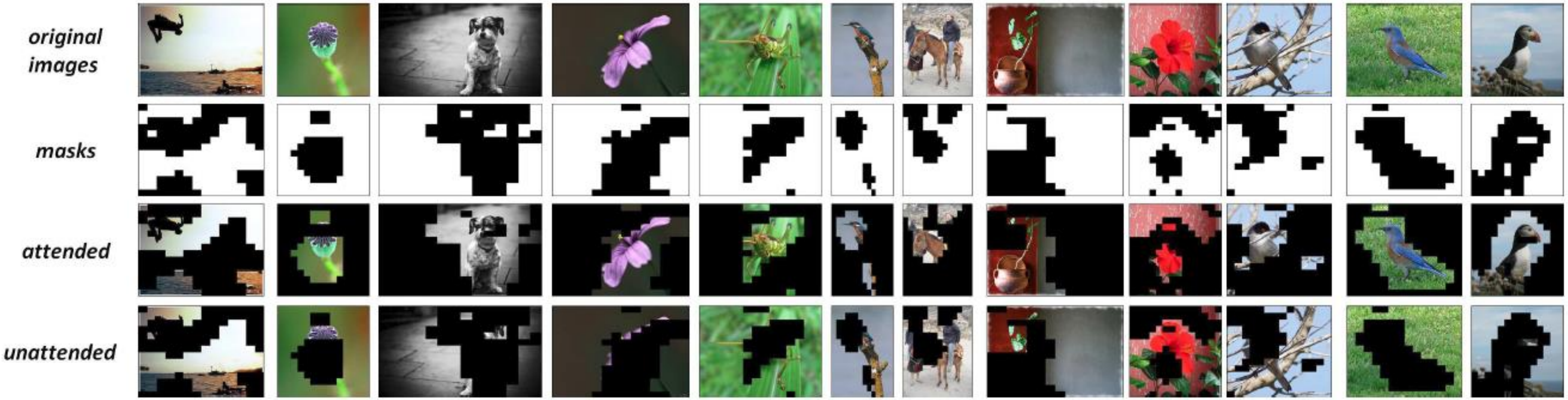
Task	Methods	16 bits	32 bits	64 bits
Text→Image	DVSH	0.6037	0.6395	0.6806
	DCMH	0.6594	0.6744	0.6905
	Ours	**0.7018**	**0.6893**	**0.6941**
Image→Text	DVSH	0.5696	0.6321	**0.6964**
	DCMH	0.5780	0.6061	0.6310
	Ours	**0.6464**	**0.6373**	0.6668

mAP with different networks

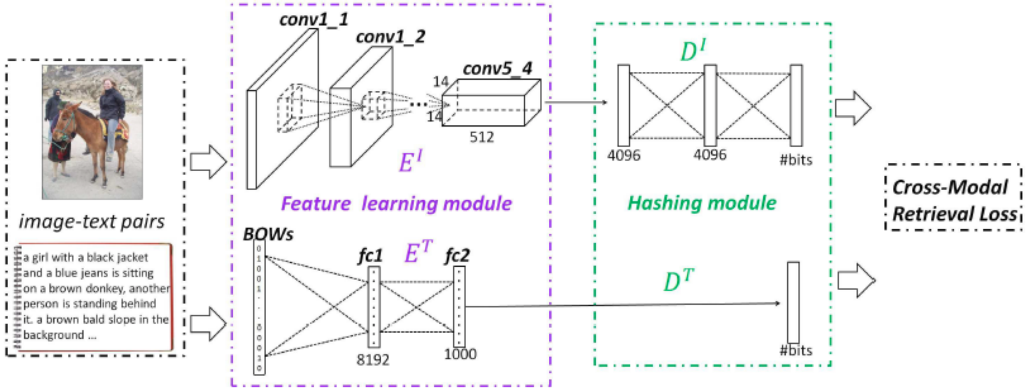
On IAPR TC-12:

Task	Networks	16 bits	32 bits	64 bits
Text→Image	VGG	0.5358	0.5565	0.5648
	CNN-F	0.5267	0.5459	0.5538
Image→Text	VGG	0.5293	0.5283	0.5439
	CNN-F	0.5211	0.5168	0.5208

Some image and mask samples

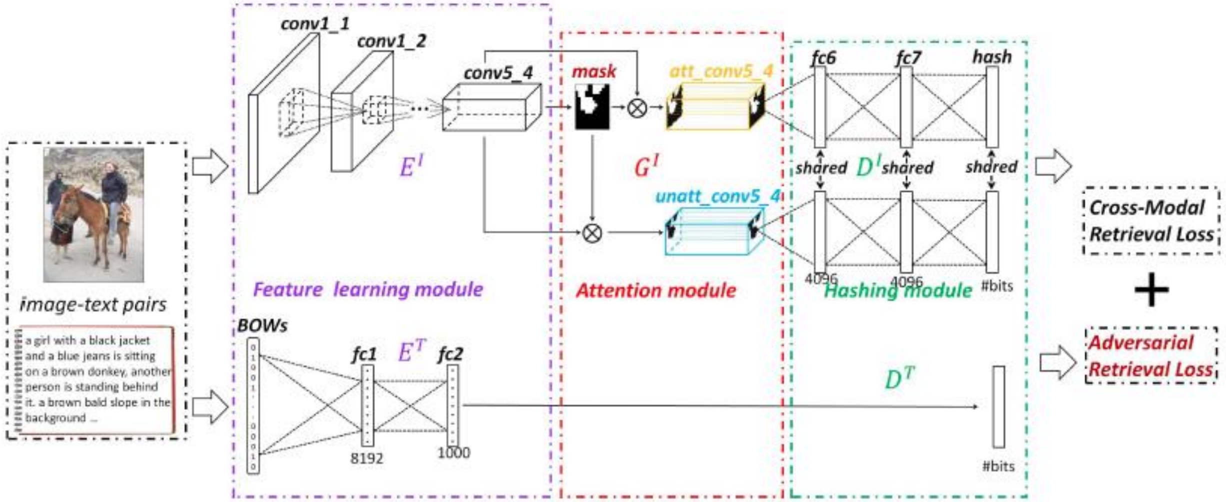


Comparison with different attention mechanisms

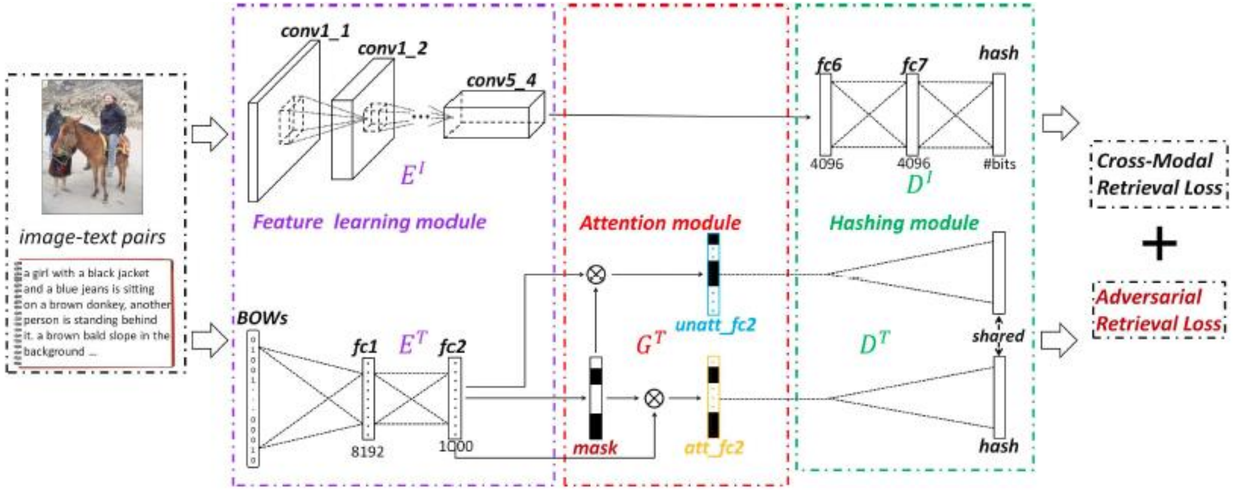


(a) No Attention

Task	Attn.	IAPR TC-12			MIR-Flickr 25k			NUS-WIDE		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
Text → Image	No	0.5039	0.5250	0.5258	0.7758	0.7801	0.7742	0.6476	0.6824	0.6733
	Visual	0.5294	0.5474	0.5576	0.7894	0.7925	0.7906	0.6723	0.6839	0.6984
	Textual	0.5334	0.5382	0.5469	0.7885	0.7867	0.7831	0.6648	0.6851	0.6867
	Both	**0.5358**	**0.5565**	**0.5648**	**0.7922**	**0.8062**	**0.8074**	**0.6789**	**0.6975**	**0.7039**
Image → Text	No	0.4903	0.5001	0.5175	0.7347	0.7482	0.7495	0.6150	0.6178	0.6311
	Visual	0.5267	0.5173	0.5285	0.7466	0.7601	0.7584	0.6314	0.6260	0.6425
	Textual	0.5279	0.5232	0.5304	0.7520	0.7673	0.7717	0.6384	0.6227	0.6459
	Both	**0.5293**	**0.5283**	**0.5439**	**0.7563**	**0.7719**	**0.7720**	**0.6403**	**0.6294**	**0.6520**



(b) Visual Attention



(c) Textural Attention

Conclusion

- Attention-based deep adversarial hashing:
  - Feature learning module
  - Attention module
  - Hashing module
- **The attention module and hashing module are trained in an adversarial way**.

Semi-supervised Generative Adversarial Hashing for Image Retrieval

Guan’an Wang, Qinghao Hu, Jian Cheng, Zengguang Hou

ECCV 2018

Background

- Nearest Neighbor Search (NNS):
  - Return the first k images with the smallest distance between the query one
  - Extremely expansive in terms of space and time.
- Approximate Nearest Neighbor Search (ANNS):
  - Return the nearest neighbors in a high probability with a sublinear or constant time complexity
  - Efficient computation and low memory cost
  - Tree based methods vs. Hashing methods

Background

- Binary Hashing
  - Traditional hashing methods: based on hand-crafted descriptors (SIFT, GIST, HOG)
    - Unsupervised methods: LSH, SH, ITQ, AGH, KMH, SpH, BRE, …
    - Semi-supervised methods: SSH
    - Supervised methods: MLH, KSH, SDH, …
  - Deep Hashing methods
    - Supervised methods: CNNH, NINH, DPSH, DHN, DSDH, …
    - Unsupervised methods: HashGAN, ?
    - Semi-supervised methods: SSDH, BGDH

- Problems:
  - Obtain labeled data is expensive <-> unlabeled data is always enough and free
  - SSDH and BGDH use graph structure to model unlabeled data -> construct graph model is expensive in time and space, and use batch data instead may lead to a suboptimal result

Semi-Supervised Generative Adversarial Hashing (SSGAH)

- Utilize a generative model to model unlabeled data and use triplet-wise labels as supervised information
- Unify a generative model, a discriminative model and a deep hashing model in an adversarial framework

- Dataset $\mathcal{X}$:
  - Unlabeled data $\mathcal{X}^u = \{x_i^u | i = 1, \dots, m\}$
  - Labeled data $\mathcal{X}^l = \{(x^q, x^p, x^n) | i = 1, \dots, n\}$ with triplet information
- Goal: learn a mapping function $\mathcal{B}(\cdot)$, $\mathcal{B}(x) \in \{0,1\}^k$ for $x \in \mathcal{X}$, while preserves relative semantic similarity of images in $\mathcal{X}$

Semi-Supervised Generative Adversarial Hashing (SSGAH)

- Generative and Discriminative models
  - Goal: learn the discrimination of unlabeled data and labeled data, and then synthesize realistic meaningful triplets
  - Given a real sample $x \in \{\mathcal{X}^u, \mathcal{X}^l\}$, generate a synthetic triplet $\{x, x_{syn}^p, x_{syn}^n\}$ where x is more similar to x_{syn}^p than to x_{syn}^n, and both synthetic ones are realistic.
  - Conditions: images -> feature v->independent Guassian distribution $N(\mu(v), \Sigma(v))$
  - Generation Discrimination Loss:

$$\begin{aligned}\min_{G} \max_{D} \mathcal{L}_{GD} = & E_{(x^q, x^p, x^n) \in \mathcal{X}^l}\{log D(x^q, x^p, x^n) + log[1 - D(x^q, x^n, x^p)]\} \\ & + E_{x \in \{\mathcal{X}^u, \mathcal{X}^l\}}\{log[1 - D(x, G_p(x), G_n(x))]\} \\ & + D_{KL}(\mathcal{N}(\mu(\nu), \Sigma(\nu)) \,||\, \mathcal{N}(0, I))\end{aligned}$$

$$KL(\mu_1, \sigma_1) = -log\sigma_1 + \frac{\sigma_1^2 + \mu_1^2}{2} - \frac{1}{2}$$

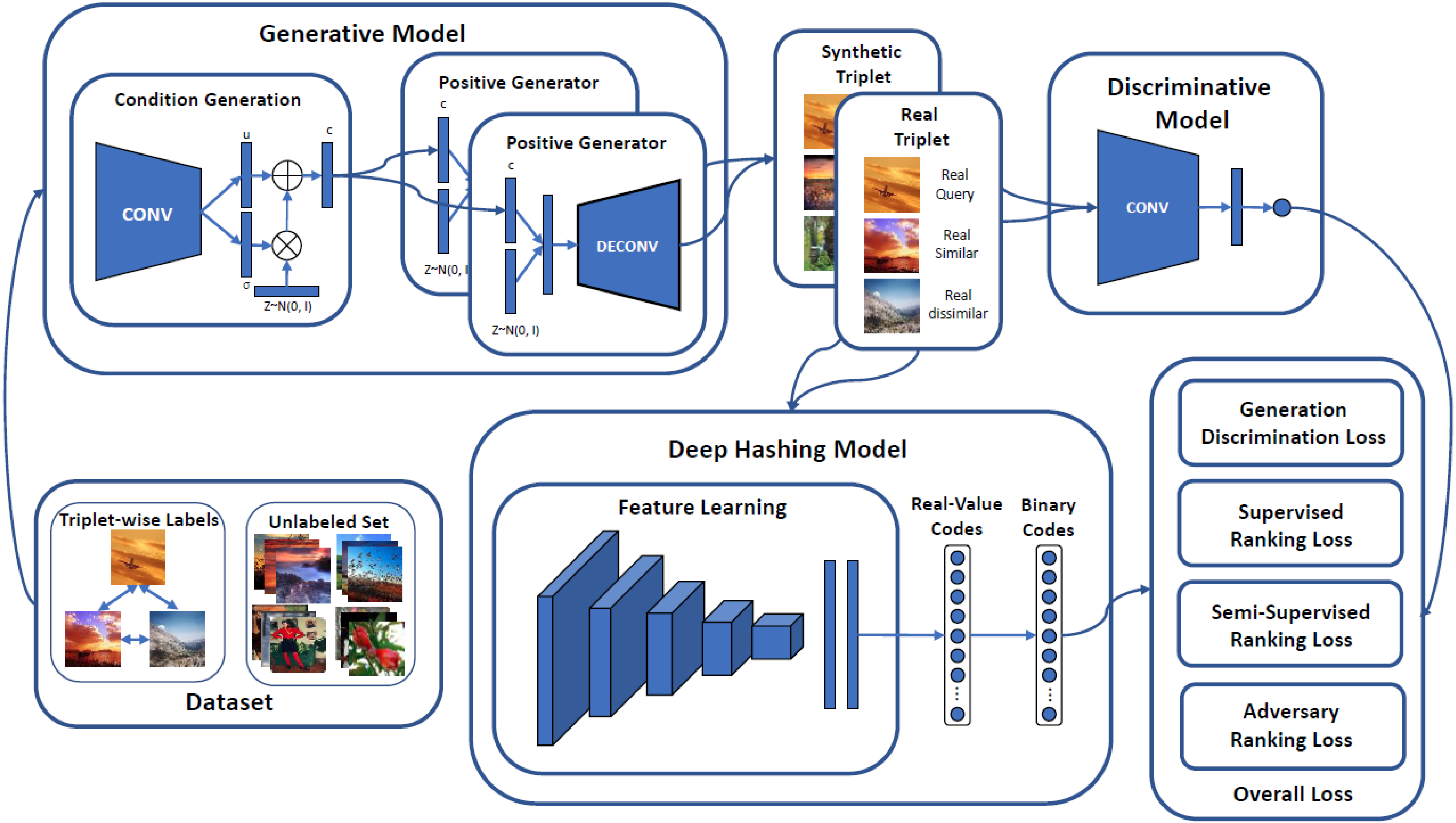
Semi-Supervised Generative Adversarial Hashing (SSGAH)

- Deep hashing model: AlexNet

$$\mathcal{H}(x) = \sigma(f(x)W^h + b^h)$$

$$\mathcal{B}(x) = (sgn(\mathcal{H}(x) - 0.5) + 1)/2$$

Model Architecture



Objective Function

- Supervised Ranking Loss:

$$\min_H \hat{\mathcal{L}}_{sr} = \sum_{i=1}^{n} \hat{\mathcal{L}}_{triplet}(m_{sr}, (x^q, x^p, x^n)_i)$$
$$= \sum_{i=1}^{n} max(0, m_{sr} - (||\mathcal{B}(x^q) - \mathcal{B}(x^n)||_H - ||\mathcal{B}(x^q) - \mathcal{B}(x^p)||_H)_i)$$

- **Semi-supervised Ranking Loss**:

$$\min_H \hat{\mathcal{L}}_{ssr} = \sum_{i=1}^{n} [\hat{\mathcal{L}}_{triplet}(m_{ssr}, (x^q, x^p_{syn}, x^n)_i) + \hat{\mathcal{L}}_{triplet}(m_{ssr}, (x^q, x^p, x^n_{syn})_i)]$$
$$+ \sum_{i=1}^{m} \hat{\mathcal{L}}_{triplet}(m_{ssr}, (x^u, x^p_{syn}, x^n_{syn})_i)$$

Objective Function

- **Adversary Ranking Loss**: minimax two-player game between the generative and deep hashing models
  - Deep hashing mode try to learn binary codes that can identify small difference between (x, x^p) and (x, x^p_{syn})
  - The generative model try to make the binary codes of x, x^p, and x^p_{syn} distinguishable

$$\min_{H} \max_{G} \hat{\mathcal{L}}_{ar} = \sum_{i=1}^{n} \hat{\mathcal{L}}_{triplet}(m_{ar}, (x^q, x^p, x^p_{syn}))$$

- Overall Objective: $\min_{G} \max_{D,H} \hat{\mathcal{L}} = \mathcal{L}_{GD} - \hat{\mathcal{L}}_{sr} - \hat{\mathcal{L}}_{ssr} - \hat{\mathcal{L}}_{ar}$

Experiments

- CIFAR-10: 60,000 32x32 color images in 10 categories
- NUS-WIDE: nearly 270,000 images with 81 concepts, select images with 21 most frequent concepts
- Query set: 100 images per class
- Training set: 500 images per class as labeled data, the others as unlabeled data

Experiment Results

mAP:

Methods	CIFAR-10				NUS-WIDE			
	12bits	24bits	32bits	48bits	12bits	24bits	32bits	48bits
SSGAH(*Ours*)	**0.819**	**0.837**	**0.847**	**0.855**	**0.835**	**0.847**	**0.859**	**0.865**
BGDH	0.805	0.824	0.826	0.833	0.803	0.818	0.822	0.828
SSDH	0.801	0.813	0.812	0.814	0.773	0.779	0.778	0.778
DSH-GANs	0.745	0.789	0.793	0.811	0.807	0.820	0.831	0.834
NINH	0.535	0.552	0.566	0.558	0.581	0.674	0.697	0.713
CNNH	0.439	0.476	0.472	0.489	0.611	0.618	0.625	0.608
SDH+CNN	0.363	0.528	0.529	0.542	0.520	0.507	0.591	0.610
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
SDH	0.255	0.330	0.344	0.360	0.414	0.465	0.451	0.454
ITQ	0.162	0.169	0.172	0.175	0.452	0.468	0.472	0.477
SH	0.124	0.125	0.125	0.126	0.433	0.426	0.426	0.423
LSH	0.116	0.121	0.124	0.131	0.404	0.421	0.426	0.441

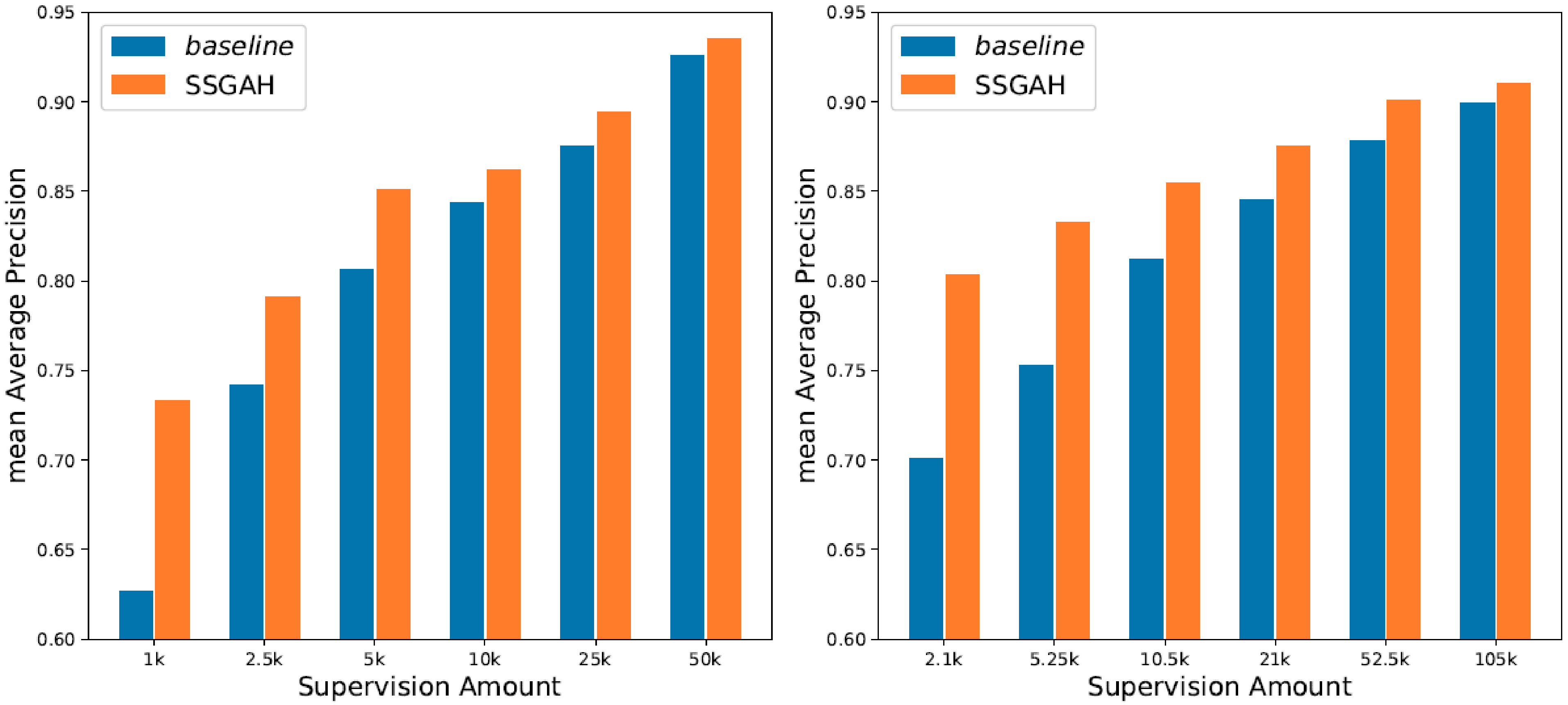
Component Analysis

- Baseline: only train H under the supervised ranking loss L_{sr}
- w/ar: train G, D and H together, but remove the semi-supervised ranking loss
- w/ssr: train G and D together under L_{GD}, and then train H under supervised ranking loss and semi-supervised ranking loss
- mAP:

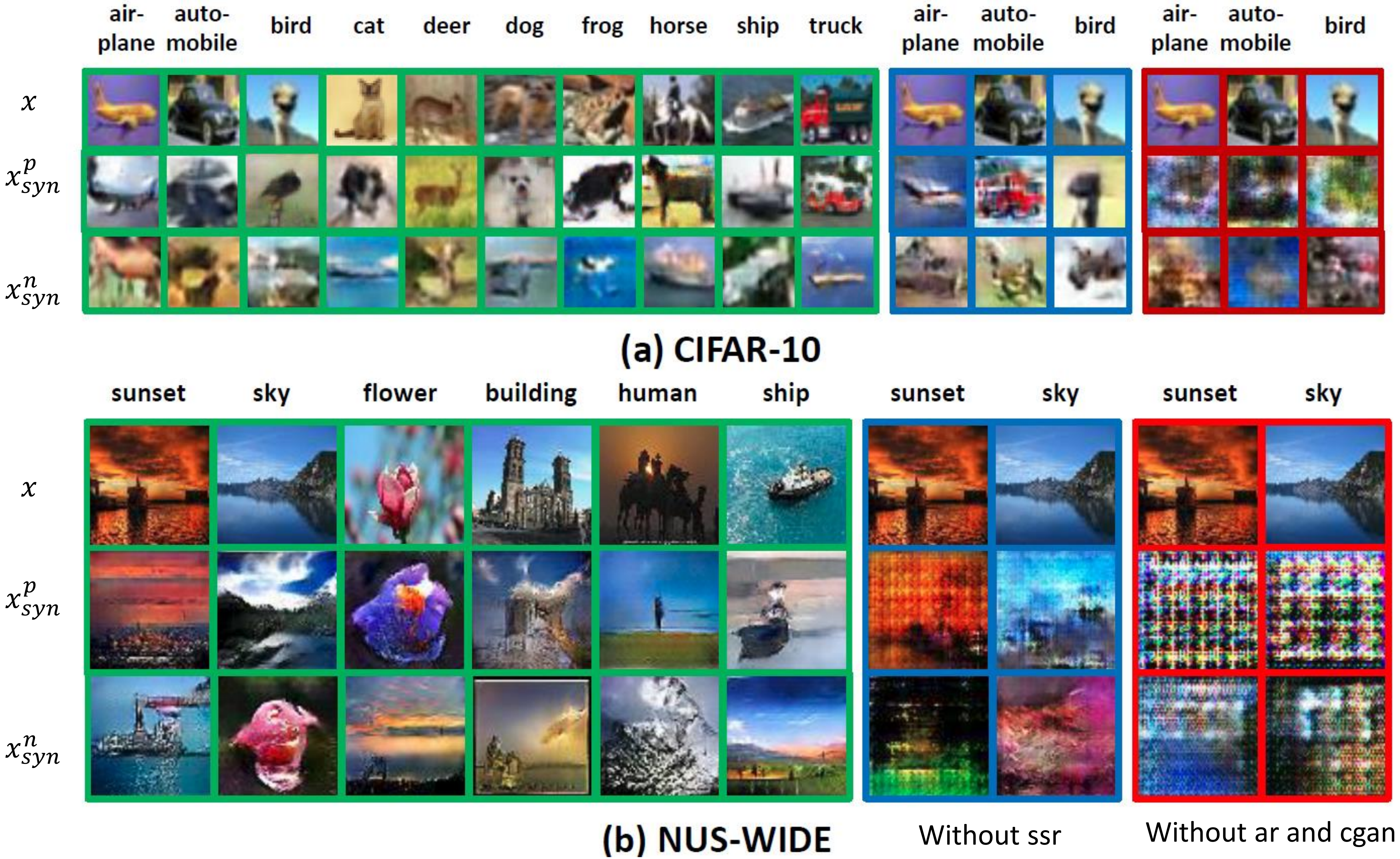
Methods	CIFAR-10				NUS-WIDE			
	12bits	24bits	32bits	48bits	12bits	24bits	32bits	48bits
SSGAH	**0.819**	**0.837**	**0.847**	**0.855**	**0.835**	**0.847**	**0.859**	**0.865**
w/ ssr	0.799	0.819	0.836	0.846	0.810	0.819	0.834	0.835
w/ ar	0.776	0.804	0.820	0.829	0.787	0.794	0.810	0.812
baseline	0.744	0.771	0.782	0.789	0.759	0.780	0.794	0.803

Effect of supervision amounts

- mAP @48 bits on CIFAR10 (left) and NUS-WIDE(right)



Visualization of Synthetic Images



Conclusion

- Semi-supervised generative adversarial hashing (SSGAH)
- Semi-supervised ranking loss and adversary ranking loss