Hugs Are Better Than Handshakes: Unsupervised Cross-Modal Transformer Hashing with Multi-granularity Alignment

Jinpeng Wang[1,3], Ziyun Zeng[1,3], Bin Chen[2,3], Yuting Wang[1,3], Dongliang Liao[4], Gongfu Li[4], Yiru Wang[4], Shu-Tao Xia[1,3]

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University [2]Harbin Institute of Technology, Shenzhen

[3]Research Center of Artificial Intelligence, Peng Cheng Laboratory [4]Wechat Group, Tencent Inc.

BMVC 2022

Summary

- Unsupervised cross-modal hashing (UCMH) aims to map different modalities into a semantic-preserving hamming space without manual labels.
- Unlike existing approaches that encoded images and texts with CNNs and MLPs, we investigate a *holistic transformer-based* framework to generate high-quality hash codes.
- We propose ***hugging***, a *multi-granularity aligning* framework for transformer-based UCMH learning.
- We instantiate *hugging* by building a simple **HuggingHash** model with a contrastive learning objective and demonstrate its merits on three datasets.
- We also adapt several state-of-the-art UCMH methods using *hugging*, verifying its feasibility on transformer-based UCMH.

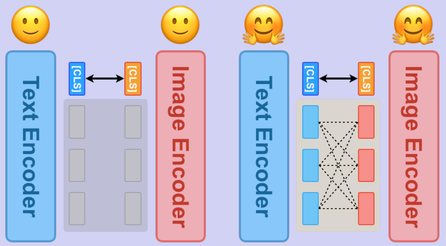
Hugging: Global + Fine-grained Alignment

- Transformer-based UCMH can be implemented easily by replacing backbones, for example

  Image Encoder: ~~VGG-16~~ → ViT

  Text Encoder: ~~MLP~~ → BERT

- Then we can train the model to align the hash codes from `[CLS]` tokens. ▷***Handshaking*** 🤝
- Handshaking is effective as expected, but it can be further enhanced to bridge the modality gap.



(a) Handshaking (b) Hugging

- **Intuition**: Content tokens that can capture fine-grained and structural modality knowledge.
- **Solution**: augment training with fine-grained alignment using content tokens.▷***Hugging*** 🤗
- **Notes**: Fine-grained alignment is an *auxiliary task* and is *removed after finishing training*. Thus, it will *not* increase extra inference overhead.

HuggingHash: A simple instantiation of hugging

- **Global Alignment**: a contrastive learning objective

$$\mathcal{L}_{\mathrm{GA}} = \frac{1}{2|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left(\frac{\exp(M_{ii}/\tau)}{\sum_{j=1}^{|\mathcal{B}|}\exp(M_{ij}/\tau)} + \frac{\exp(M_{ii}/\tau)}{\sum_{j=1}^{|\mathcal{B}|}\exp(M_{ji}/\tau)}\right)$$

cosine similarity between hash codes $M_{ij} = \cos(\boldsymbol{b}_i, \boldsymbol{b}_j)$

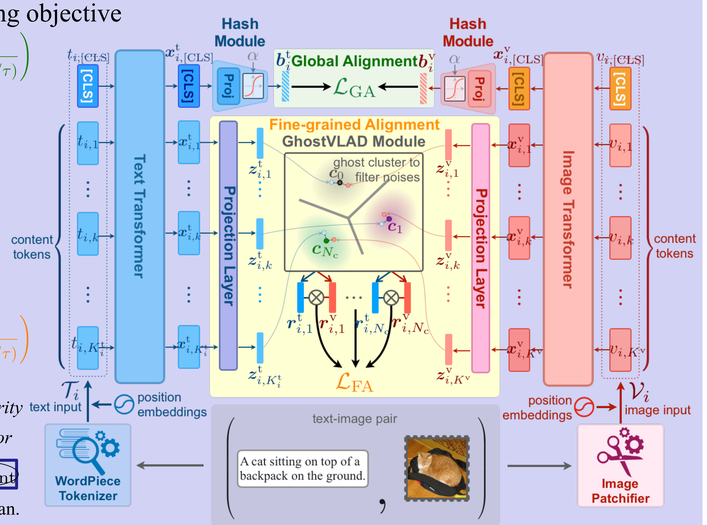
- **Fine-grained Alignment**:
  1. Using GhostVLAD[1] to transform content tokens into a fixed number of fine-grained embeddings;
  2. Cluster-wised contrastive learning

$$\mathcal{L}_{\mathrm{FA}} = \frac{1}{N_c|\mathcal{B}|}\sum_{n=1}^{N_c}\sum_{i=1}^{|\mathcal{B}|}\left(\frac{\exp(m_{ii}^n/\tau)}{\sum_{j=1}^{|\mathcal{B}|}\exp(m_{ij}^n/\tau)} + \frac{\exp(m_{ii}^n/\tau)}{\sum_{j=1}^{|\mathcal{B}|}\exp(m_{ji}^n/\tau)}\right)$$

number of clusters/fine-grained embeddings $m_{ij}^n = \cos(\boldsymbol{r}_{i,n}^{\mathrm{t}}, \boldsymbol{r}_{j,n}^{\mathrm{v}})$ *the n-th fine-grained cosine similarity*

- **Total Learning Objectives**: *quantization error*

$$\mathcal{L}_{\mathrm{HuggingHash}} = \mathcal{L}_{\mathrm{GA}} + \lambda\mathcal{L}_{\mathrm{FA}} + \gamma\mathcal{R}_{\mathrm{quant}}$$



[1] Yujie Zhong, Relja Arandjelović, and Andrew Zisserman. Ghostvlad for set-based face recognition. In *ACCV'18*.

Results and Analyses

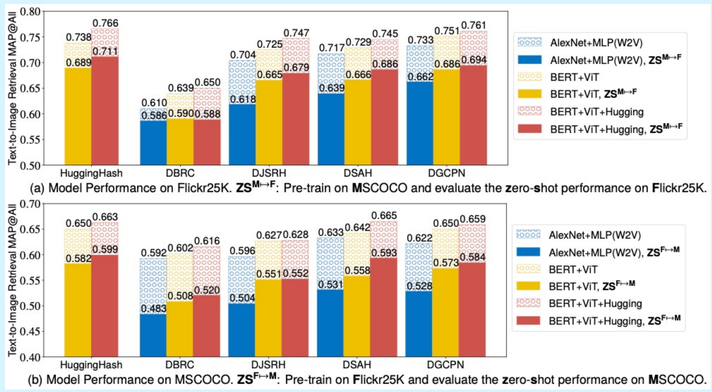
- **Part 1: Compare HuggingHash with UCMH SOTAs**

Table 1. *Cross-modal retrieval mean average precision (MAP) results for different numbers of bits on the three datasets. 'I2T' and 'T2I' stand for 'Image-to-Text' and 'Text-to-Image' for short, respectively. The best value of each column is shown in boldface.*

Method	Venue	Backbone	Flickr25K T2I 16 bits	Flickr25K T2I 32 bits	Flickr25K T2I 64 bits	Flickr25K I2T 16 bits	Flickr25K I2T 32 bits	Flickr25K I2T 64 bits	MSCOCO T2I 16 bits	MSCOCO T2I 32 bits	MSCOCO T2I 64 bits	MSCOCO I2T 16 bits	MSCOCO I2T 32 bits	MSCOCO I2T 64 bits	Wiki T2I 16 bits	Wiki T2I 32 bits	Wiki T2I 64 bits	Wiki I2T 16 bits	Wiki I2T 32 bits	Wiki I2T 64 bits
CVH [24]	IJCAI'11	Non-Deep	0.607	0.591	0.581	0.602	0.587	0.578	0.507	0.479	0.446	0.499	0.471	0.443	0.252	0.235	0.171	0.179	0.162	0.153
IMH [49]	SIGMOD'13		0.586	0.593	0.589	0.588	0.581	0.585	0.413	0.435	0.443	0.416	0.435	0.443	0.467	0.478	0.453	0.201	0.203	0.204
CMFH [58]	TIP'16		0.611	0.606	0.575	0.659	0.660	0.663	0.453	0.435	0.499	0.442	0.423	0.492	0.595	0.601	0.616	0.251	0.253	0.259
FSH [32]	CVPR'17		0.589	0.595	0.595	0.590	0.597	0.597	-	-	-	-	-	-	-	-	-	-	-	-
ACQ [22]	ICCV'17		-	-	-	-	-	-	0.565	0.561	0.520	0.559	0.553	0.515	-	-	-	-	-	-
DBRC [30]	MM'17	CNN + MLP	0.591	0.596	0.598	0.596	0.600	0.602	0.562	0.566	0.573	0.555	0.561	0.564	0.574	0.588	0.598	0.253	0.265	0.269
UGACH [67]	AAAI'18		0.676	0.692	0.703	0.676	0.693	0.702	0.566	0.595	0.607	0.550	0.584	0.599	0.544	0.555	0.572	0.388	0.392	0.403
UCH [26]	AAAI'19		0.661	0.668	0.654	0.669	0.679	0.686	0.446	0.469	0.488	0.447	0.471	0.485	-	-	-	-	-	-
DJSRH [50]	ICCV'19		0.683	0.694	0.717	0.666	0.678	0.699	0.573	0.578	0.584	0.572	0.575	0.579	0.611	0.635	0.646	0.388	0.403	0.412
UKD-SS [18]	CVPR'20		0.704	0.705	0.714	0.700	0.706	0.709	0.580	0.594	0.603	0.564	0.592	0.601	0.556	0.565	0.578	0.403	0.411	0.416
SRCH [56]	IJCAI'20		-	-	-	-	-	-	0.600	0.606	0.623	0.598	0.605	0.623	-	-	-	-	-	-
DSAH [60]	MM'20		0.707	0.713	0.728	0.701	0.712	0.722	0.606	0.599	0.620	0.598	0.589	0.609	0.644	0.650	0.660	0.416	0.430	0.438
DGCPN [63]	AAAI'21		0.729	0.741	0.749	0.732	0.742	0.751	0.594	0.603	0.616	0.587	0.594	0.612	0.629	0.638	0.641	0.422	0.440	0.446
DBRC [30]	MM'17	Transformers	0.628	0.633	0.639	0.627	0.637	0.642	0.592	0.605	0.602	0.594	0.603	0.611	0.591	0.594	0.599	0.449	0.460	0.466
DJSRH [50]	ICCV'19		0.716	0.724	0.725	0.712	0.718	0.723	0.623	0.621	0.627	0.619	0.624	0.627	0.640	0.649	0.652	0.496	0.502	0.511
DSAH [60]	MM'20		0.726	0.729	0.729	0.723	0.727	0.734	0.641	0.636	0.642	0.637	0.639	0.648	0.655	0.661	0.667	0.491	0.489	0.501
DGCPN [63]	AAAI'21		0.743	0.756	0.751	0.745	0.750	0.755	0.638	0.649	0.650	0.633	0.641	0.643	0.651	0.658	0.662	0.489	0.498	0.503
HuggingHash	BMVC'22		**0.745**	**0.760**	**0.766**	**0.752**	**0.758**	**0.764**	**0.652**	**0.661**	**0.663**	**0.646**	**0.653**	**0.662**	**0.659**	**0.669**	**0.675**	**0.522**	**0.520**	**0.526**

Analysis: In terms of global alignment for hash codes, though HuggingHash adopts a contrastive learning objective that is much simpler than the baselines, the superior results indicate the effectiveness of exploring *multi-granularity alignment* with transformers beyond *global alignment* itself.

- **Part 2: Model Transferability**



(a) Model Performance on Flickr25K. $\mathbf{ZS}^{\mathbf{M}\to\mathbf{F}}$: Pre-train on **M**SCOCO and evaluate the zero-**s**hot performance on **F**lickr25K.

(b) Model Performance on MSCOCO. $\mathbf{ZS}^{\mathbf{F}\to\mathbf{M}}$: Pre-train on **F**lickr25K and evaluate the zero-**s**hot performance on **M**SCOCO.

Figure 1. *Multi-dataset (Flickr25K-MSCOCO) evaluation with different 64-bit UCMH methods.*

Analyses: Transformers and the hugging improve generalizability and robustness. Besides, HuggingHash shows best zero-shot results.

- **Part 3: Enhancing UCMH SOTAs with *hugging***

Table 2. *Retrieval mean average precision (MAP@All) results on MSCOCO. `Use Tr' means whether the variant uses transformers.*

Method	Use Tr *Hugging*		T2I Retrieval 16 bits	T2I Retrieval 32 bits	T2I Retrieval 64 bits	I2T Retrieval 16 bits	I2T Retrieval 32 bits	I2T Retrieval 64 bits
DBRC [30]			0.562	0.566	0.573	0.555	0.561	0.564
	✓		0.592	0.605	0.602	0.594	0.603	0.611
	✓	✓	0.611	0.613	0.616	0.610	0.619	0.630
DJSRH [50]			0.573	0.578	0.584	0.572	0.575	0.579
	✓		0.623	0.621	0.627	0.619	0.624	0.627
	✓	✓	0.637	0.626	0.628	0.630	0.634	0.637
DSAH [60]			0.606	0.599	0.620	0.598	0.589	0.609
	✓		0.641	0.636	0.642	0.637	0.639	0.648
	✓	✓	0.656	0.659	0.665	0.669	0.663	0.670
DGCPN [63]			0.594	0.603	0.616	0.587	0.594	0.612
	✓		0.638	0.649	0.650	0.633	0.641	0.643
	✓	✓	0.645	0.662	0.659	0.644	0.650	0.658
HuggingHash	✓		0.633	0.644	0.650	0.631	0.636	0.644
	✓	✓	0.652	0.661	0.663	0.646	0.653	0.662

Analyses: We can see consistent improvements. Besides, SOTA methods surpass HuggingHash when equipped with hugging. This is reasonable since they use more complex global alignment mechanisms.