Supplementary Material for "Object-Aware Query Perturbation for Cross-Modal Image Retrieval"

Naoya Sogi, Takashi Shibata, and Makoto Terao

Visual Intelligence Research Laboratories, NEC Corporation, Kanagawa, Japan
naoya-sogi@nec.com, t.shibata@ieee.org, m-terao@nec.com

A Experimental details

A.1 Evaluation Metrics

As with the previous works, we used Recall@K, which is a ratio of correct retrievals to all retrievals. For the text-to-image (T2I) retrieval task, the correct retrieval is identified by whether the image corresponding to the input text is in the top K retrieval results. For the image-to-text (I2T) retrieval task, the correct retrieval is identified by whether at least one of the five description texts of the input image is in the top K retrieval results.

Our mean Recall@K is a harmonic mean of multiple Recall@Ks; each of them is calculated with the above procedure on a subset of test data. We used ten subsets by splitting the largest object areas in each image by every 10%.

A.2 Implementation Details

Q-Perturbation. We applied our Q-Perturbation to the following cross-attention modules: for BLIP2, six cross-attentions in Q-Former; for InternVL, sixteen cross-attentions in QLLaMA and one additional cross-attention after vision encoder; for COCA, one cross attention after vision encoder. For BLIP2, Q- Perturbation was applied for the image-text contrastive (ITC) phase.

For multi-head cross-attention modules, Q-Perturbation is applied to each head. Algorithm 1.1 shows a PyTorch-like pseudo code of Q-Perturbation for each head of cross attentions. This function is inserted before calculating attention scores.

To implement image captioning by BLIP2 with Q-Perturbation, we used the language head following Q-Former [2].

Generation of a Key Subspace. Let $\mathbf{K}^{obj} \in \mathbb{R}^{D \times N}$ be a matrix, whose columns are image tokens selected by our Object Key Pooling; each column is a key vector in a cross-attention overlapping a detected object. Here, D is the dimension of a vector, and N is the number of tokens.

Algorithm 1.1: PyTorch-like pseudo code for our Q-Perturbation.

```python

def query_perturbation(
    q: torch.Tensor,
    k: torch.Tensor,
    obj_mask: torch.Tensor,
    alpha: torch.Tensor,
    contrib_th: float,
):
    """Query-Perturbation.

    Apply query perturbation to the queries using the keys and object masks.

    Args:
        q (torch.Tensor): Queries. Shape: [#queries, dims]
        k (torch.Tensor): Keys. Shape: [#keys, dims]
        obj_mask (torch.Tensor): Object masks. Shape: [#objects, #keys]
        alpha (torch.Tensor): Weight scale. Shape: [#objects]
        contrib_th (float): Threshold for PCA.

    Returns:
        torch.Tensor: Perturbed queries. Shape: [#queries, dims]
    """
    pert_vec = torch.zeros_like(q)
    for i in range(obj_mask.shape[0]):
        # 1. Extract the keys corresponding to the object
        k_obj = k * obj_mask[i].unsqueeze(1)

        # 2. Make the key subspace; Apply PCA to the keys
        phi = PCA(k_obj, contrib_th)  # phi = [#n_components, dims]
        proj_mat = phi.T @ phi  # proj_mat = [dims, dims]

        # 3. Decompose the queries with the key subspace
        q_proj = q @ proj_mat  # q_proj = [#queries, dims]
        pert_vec += alpha[i] * q_proj

    # 4. perturb the queries
    return q + pert_vec
```

We generate basis vectors $\{\phi_i\}$ of a key subspace by applying Principal Component Analysis (PCA) to $\mathbf{K^{obj}}$. Formally, basis vectors are obtained by solving the following eigenvalue decomposition problem:

$$\mathbf{K^{obj}}\mathbf{K^{obj}}^T = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Phi}^T, \tag{1}$$

where, diagonal elements $\sigma_{i,i}$ of $\mathbf{\Sigma}$ are eigenvalues ordering descending order, i.e., $\sigma_{1,1} > \sigma_{2,2,} > ... > \sigma_{min(D,N),min(D,N)}$, and eigenvectors $\mathbf{\Phi} = [\phi_1, \phi_2, ..., \phi_{min(D,N)}]$ are candidates for basis vectors of a key subspace. We select the first p eigenvectors, $[\phi_1, \phi_2, ..., \phi_p]$, as basis vectors. The number of basis vectors p is set based on the following contribution ratio:

$$c_p = \frac{\sum_{i=1}^{p} \sigma_{i,i}}{\sum_{j=1}^{min(D,N)} \sigma_{j,j}}. \tag{2}$$

Concretely, the number of basis vectors p is set at the maximum value, while ensuring that the contribution ratio does not exceed a threshold. In the experiments, the threshold is selected by the grid-search algorithm.

Table A: Comparative results on Flickr30K for the T2I task with two baselines, OEs and OrQ-Pert.

	10%	R@1	R@5	mR@1	mR@5
BLIP2	81.33	89.76	98.18	88.75	97.85
w/Q-Pert. (E)	**84.00**	**89.82**	**98.20**	**89.10**	**97.89**
w/OEs	56.00	78.48	94.84	74.28	94.10
w/OrQ-Pert.	77.33	89.64	98.12	88.16	97.53

A.3 Other Details

All the experiments were conducted using computation servers equipped with NVIDIA's GeForce GTX 1080Ti and Tesla A100. We used PyTorch [3] to implement the experiments.

B Additional Experimental Results

B.1 Ablation Studies

Comparison with an Object-Aware Baseline We evaluated an another object-aware baseline regarding [1], as shown in Table A. This baseline uses the sum of the original BLIP2s' score and the highest similarity between text and object embeddings (OEs) for retrieval. OEs are obtained by applying ROI pooling to the visual tokens from the BLIP2s' backbone and parsing the pooled tokens to Q-Former. We could see the effectiveness of our method again, as our method outperforms this simple baseline.

Performance Analysis of Query Enhancement We evaluated the modified Q-Pert. (OrQ-Pert.) to confirm the effectivenss of our method. Orthogonal Q-pert. (OrQ-Pert.) uses the projected query into an orthogonal complement of K-subspace instead of our perturbed query, i.e., it reduces object awareness. As shown in Table A, OrQ-Pert. shows lower performances, especially on small objects. The result supports the effectiveness of our method.

B.2 Computational Cost

Table B shows the extra computational cost. The extra costs are not dominant. The main bottleneck is the visual backbone, which is also the same as COCA and InternVL. It would be an excellent future direction to consider an algorithm for pruning target cross-attention layers of our method to reduce the extra cost further. Note that image features can be pre-computed for text-to-image retrieval.

Table B: Computational time [milliseconds] of visual feature extraction by VLMs. Q-Pert.(KC) and (QE) show the extra costs by K-subspace Construction and Query-Enhancement.

GPU	BLIP2	w/Q-Pert.	Q-Pert.(KC)	Q-Pert.(QE)
GTX 1080Ti	321.0	413.0	+74.9	+1.3
RTX 8000	128.0	187.0	+57.1	+0.6

Table C: Sensitivity on hyperparameters, i.e., the weight function, and scale factor, with two evaluation indexes.

(a) Flickr-30K dataset. R@1 small(10%).

	scale				
	2	4	6	8	10
$\bar{S}_b$	81.33	$\underline{82.67}$	81.33	80.00	81.33
1-$\bar{S}_b$	$\underline{82.67}$	$\underline{82.67}$	80.00	81.33	**84.00**
$\bar{S}_b$-0.5	81.33	81.33	81.33	$\underline{82.67}$	$\underline{82.67}$
0.5-$\bar{S}_b$	81.33	$\underline{82.67}$	81.33	81.33	81.33
constant	$\underline{82.67}$	81.33	81.33	81.33	**84.00**

(b) Flickr-30K dataset. R@1.

	scale				
	2	4	6	8	10
$\bar{S}_b$	88.73	88.65	88.67	88.67	88.77
1-$\bar{S}_b$	88.85	88.63	88.63	$\underline{88.93}$	88.62
$\bar{S}_b$-0.5	88.69	88.41	88.75	88.46	88.52
0.5-$\bar{S}_b$	88.53	88.80	88.81	88.72	88.63
constant	88.63	**88.95**	$\underline{88.93}$	88.67	88.76

Table D: Sensitivity on hyperparameters, i.e., the weight function, and dimension of subspace, with two evaluation indexes.

(a) Flickr-30K dataset. R@1 small(10%).

	Contribution ratio				
	0.80	0.85	0.90	0.95	0.99
$\bar{S}_b$	81.33	81.33	81.33	81.33	82.67
1-$\bar{S}_b$	81.33	$\underline{84.00}$	$\underline{84.00}$	82.67	81.33
$\bar{S}_b$-0.5	81.33	80.00	80.00	80.00	81.33
0.5-$\bar{S}_b$	81.33	81.33	80.00	81.33	81.33
constant	81.33	81.33	**85.33**	82.67	$\underline{84.00}$

(b) Flickr-30K dataset. R@1.

	Contribution ratio				
	0.80	0.85	0.90	0.95	0.99
$\bar{S}_b$	88.60	88.60	88.75	88.59	88.77
1-$\bar{S}_b$	88.84	**89.16**	88.84	88.81	88.62
$\bar{S}_b$-0.5	88.83	88.46	88.43	88.49	88.52
0.5-$\bar{S}_b$	88.78	88.75	88.51	88.67	88.63
constant	88.57	88.67	$\underline{89.14}$	88.70	88.76

B.3 Sensitivity on Hyper Parameters

Table C D shows retrieval performances by our method with varying hyperparameters. We can see that the proposed method has low sensitivities to the hyperparameters.

B.4 Additional Examples

Figure A shows examples of image retrieval by BLIP2 and BLIP2 with Q-Perturbation. It seems that our method could perform detailed matching between a text and images, making accurate retrievals.

Figure B shows examples of image captioning results by BLIP2 and BLIP2 with Q-Perturbation. The results suggests that our method enables captioning by a V&L model to be controllable using object information.

References

1. Chen, B.C., Wu, Z., Davis, L.S., Lim, S.N.: Efficient Object Embedding for Spliced Image Retrieval. In: CVPR. pp. 14960–14970 (2021)
2. Li, Junnan and Li, Dongxu and Savarese, Silvio and Hoi, Steven: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: ICML. pp. 19730–19742 (2023)
3. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: NeurIPS. pp. 8024–8035 (2019)

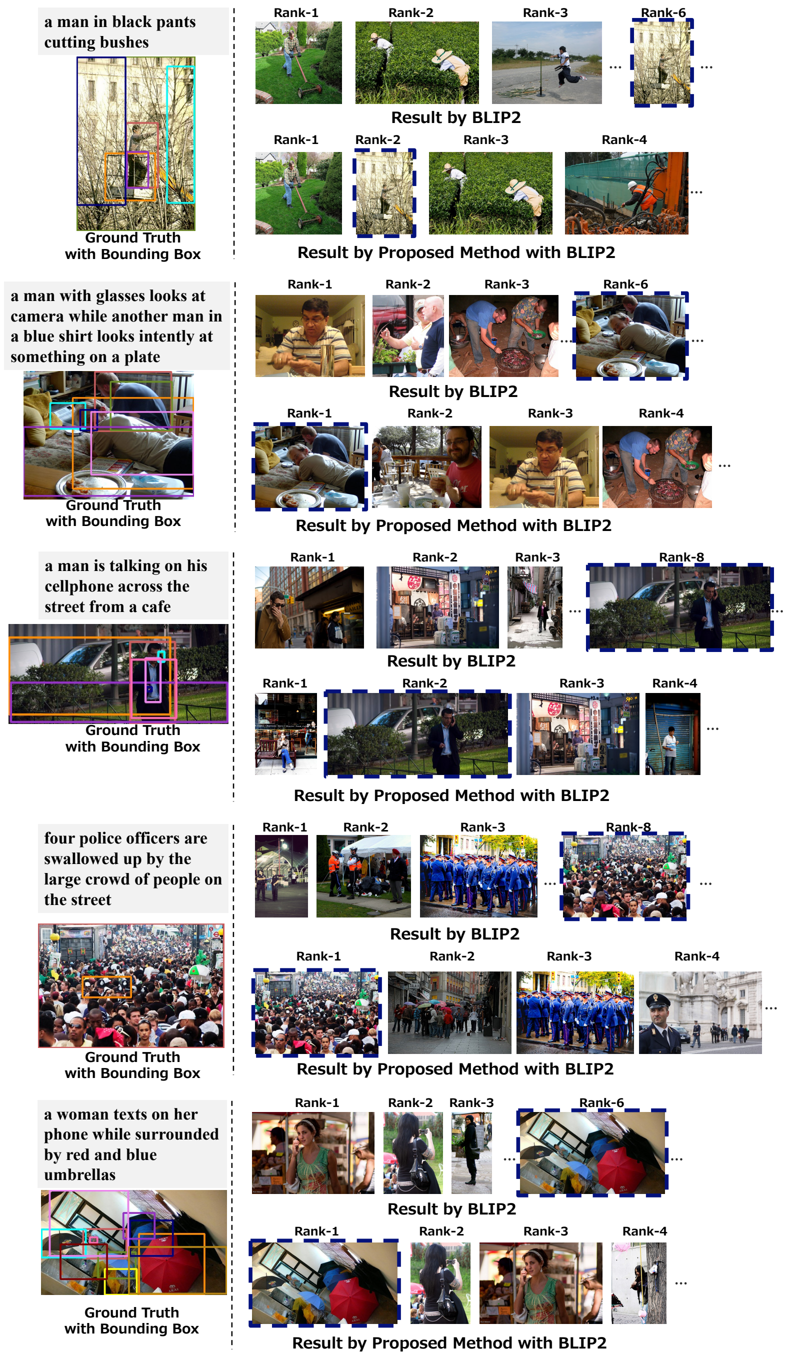
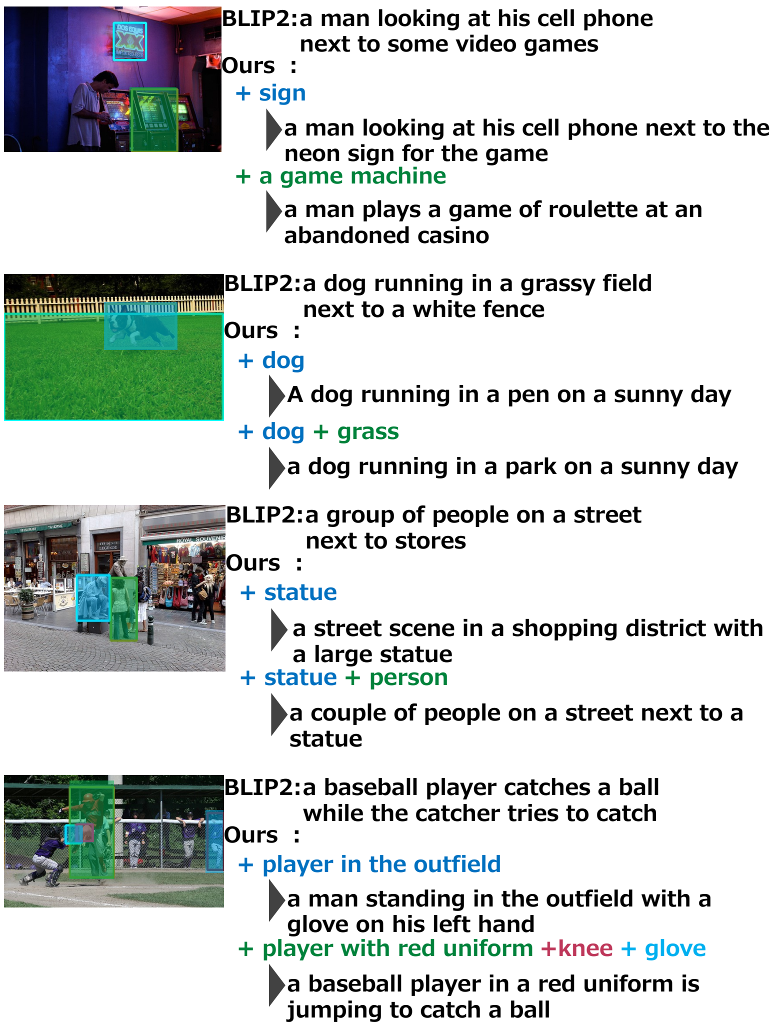

a man in black pants cutting bushes
Ground Truth with Bounding Box
Rank-1
Rank-2
Rank-3
Rank-6
Result by BLIP2
Rank-1
Rank-2
Rank-3
Rank-4
Result by Proposed Method with BLIP2
a man with glasses looks at camera while another man in a blue shirt looks intently at something on a plate
Ground Truth with Bounding Box
Rank-1
Rank-2
Rank-3
Rank-6
Result by BLIP2
Rank-1
Rank-2
Rank-3
Rank-4
Result by Proposed Method with BLIP2
a man is talking on his cellphone across the street from a cafe
Ground Truth with Bounding Box
Rank-1
Rank-2
Rank-3
Rank-8
Result by BLIP2
Rank-1
Rank-2
Rank-3
Rank-4
Result by Proposed Method with BLIP2
four police officers are swallowed up by the large crowd of people on the street
Ground Truth with Bounding Box
Rank-1
Rank-2
Rank-3
Rank-8
Result by BLIP2
Rank-1
Rank-2
Rank-3
Rank-4
Result by Proposed Method with BLIP2
a woman texts on her phone while surrounded by red and blue umbrellas
Ground Truth with Bounding Box
Rank-1
Rank-2
Rank-3
Rank-6
Result by BLIP2
Rank-1
Rank-2
Rank-3
Rank-4
Result by Proposed Method with BLIP2

Fig. A: Examples of image retrieval results by BLIP2 [2] and our method.



BLIP2:a man looking at his cell phone next to some video games
Ours :
+ sign
▶a man looking at his cell phone next to the neon sign for the game
+ a game machine
▶a man plays a game of roulette at an abandoned casino

BLIP2:a dog running in a grassy field next to a white fence
Ours :
+ dog
▶A dog running in a pen on a sunny day
+ dog + grass
▶a dog running in a park on a sunny day

BLIP2:a group of people on a street next to stores
Ours :
+ statue
▶a street scene in a shopping district with a large statue
+ statue + person
▶a couple of people on a street next to a statue

BLIP2:a baseball player catches a ball while the catcher tries to catch
Ours :
+ player in the outfield
▶a man standing in the outfield with a glove on his left hand
+ player with red uniform +knee + glove
▶a baseball player in a red uniform is jumping to catch a ball

Fig. B: Image captioning results by BLIP2 and our method. "+ word" means enhancing information of the corresponding object