HQA-VLAttack: Towards High Quality Adversarial Attack on Vision-Language Pre-Trained Models

Han Liu[1], Jiaqi Li[1], Zhi Xu[1], Xiaotong Zhang[1], Xiaoming Xu[1],
Fenglong Ma[2], Yuanman Li[3], Hong Yu[1]

[1] Dalian University of Technology [2]The Pennsylvania State University [3]Shenzhen University

Introduction

➢ Vision-Language Pre-training (VLP) models

VLP models have achieved significant development in various domains. However, recent studies have shown that **these models are vulnerable to adversarial attacks.**

➢ Previous Works

Current adversarial attack methods ignore the impact of negative image-text pairs on attack success rate. We want to propose an **efficient** and **simple** paradigm to attack VLP models.

Related Work

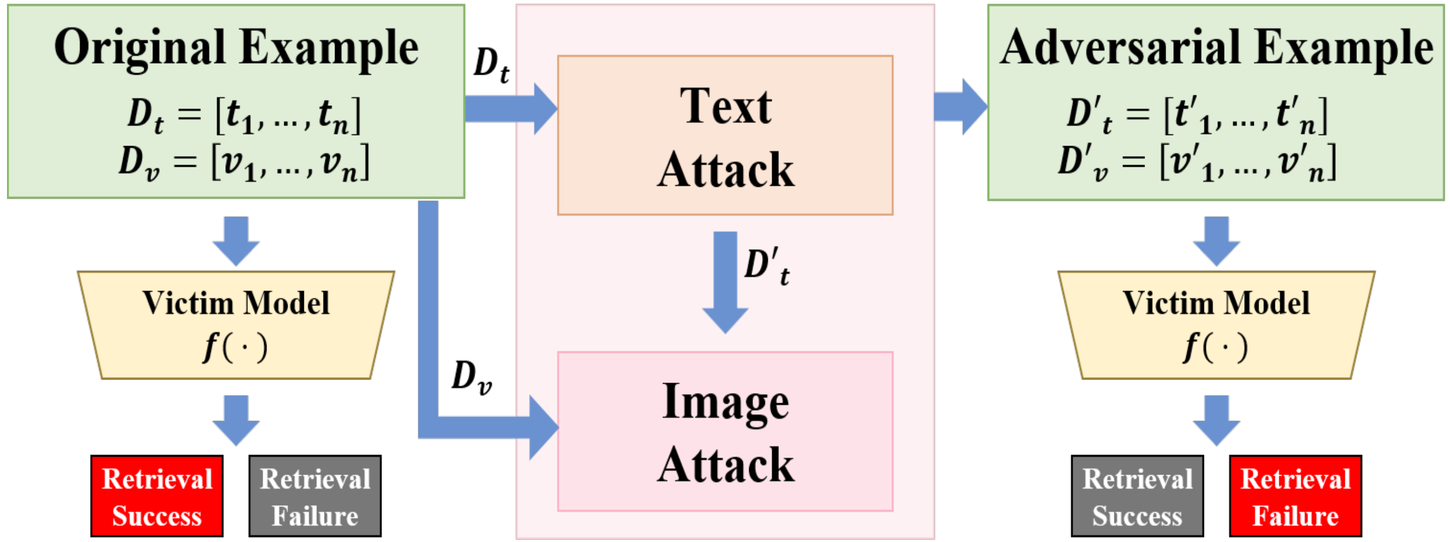
① White-Box Adversarial Attacks

- **Access**: **All information about the victim model**
- **Advantages**: Simple and efficient
- **Disadvantages**: Not practical
- **Strategy**: Co-Attack

② Black-Box Adversarial Attacks

- **Access**: **Confidence scores** or **Predicted label**
- **Advantages**: Practical
- **Disadvantages**: Complex and inefficient
- **Strategy**: VLAttack,SGA,DRA

Problem Formulation



◆ What is the adversarial example?

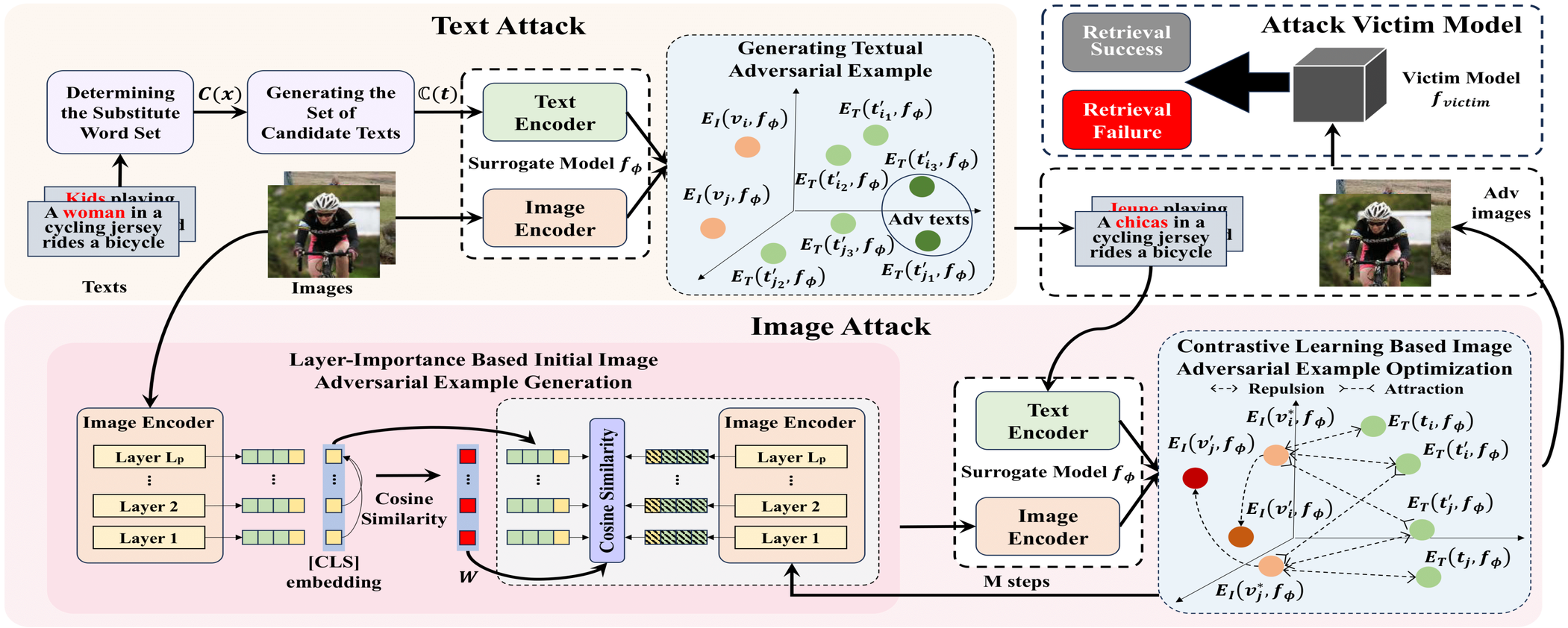
Let D_v, D_t denote original image and textual sets. By applying pixel-level perturbations to images and word-level perturbations to texts using a surrogate model f_ϕ. We can get **image and textual adversarial example sets** D'_v and $D_t{}'$, which can lead the victim model f to retrieval failure.

◆ What is the goal of multimodal adversarial example?

To find the best adversarial example, the difference from the original sample must be constrained within a certain perturbation limit.

$$v'_i \notin F_{IR}(t'_i, D'_v), \qquad t'_i \notin F_{TR}(v'_i, D'_t) \quad s.t. \ \ ||v'_i - v_i||_\infty \leq \epsilon_v \, , d(t'_i, t_i) \leq \epsilon_t.$$

HQA-VLAttack



A.Text Attack

A.1 Determining the Substitute Word Set.

A.2 Generating Textual Adversarial Example

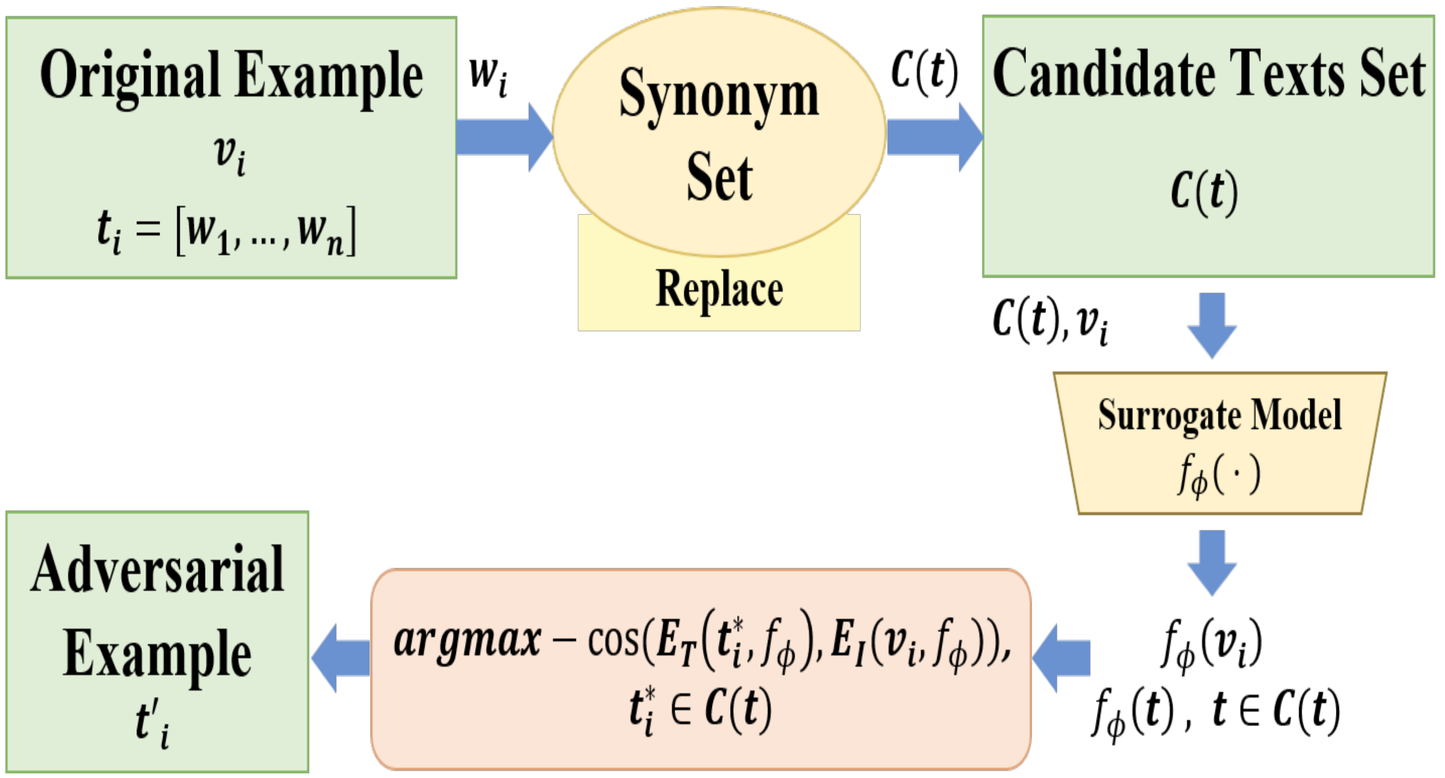
B. Image Attack

B.1 Layer-Importance Based Initial Image Adversarial Example Generation

B.2 Contrastive Learning Based Image Adversarial Example Optimization

HQA-VLAttack

➢ Text Attack



1. Determining the Substitute Word Set

To fix the semantic inconsistency of MLM-generated substitute words, we primarily use counter-fitting word vectors to find synonyms based on cosine similarity, using the BERT-Attack method as a fallback.

2. Generating Textual Adversarial Example

We generate a set of candidate texts containing all possible single-word synonym replacements and select the text with the lowest cosine similarity to the original image's features as the final adversarial sample.

HQA-VLAttack

➢ Text Attack

1 Determining the Substitute Word Set

To fix the semantic inconsistency of MLM-generated substitute words, we primarily use counter-fitting word vectors to find synonyms based on cosine similarity, using the BERT-Attack method as a fallback.

$$t_i = [x_1, \ldots, x_j, \ldots, x_L]$$

$$C(x_j) = \begin{cases} x'_j \mid \cos(\mathbf{v}_{x'_j}, \mathbf{v}_{x_j}) > \tau, & \text{if } \mathbf{v}_{x_j} \in \mathbf{V}_{cf}, \\ \text{argmax}_k f_{\text{mlm}}(x_j), & \text{otherwise}, \end{cases}$$

HQA-VLAttack

➢ Text Attack

1 Determining the Substitute Word Set

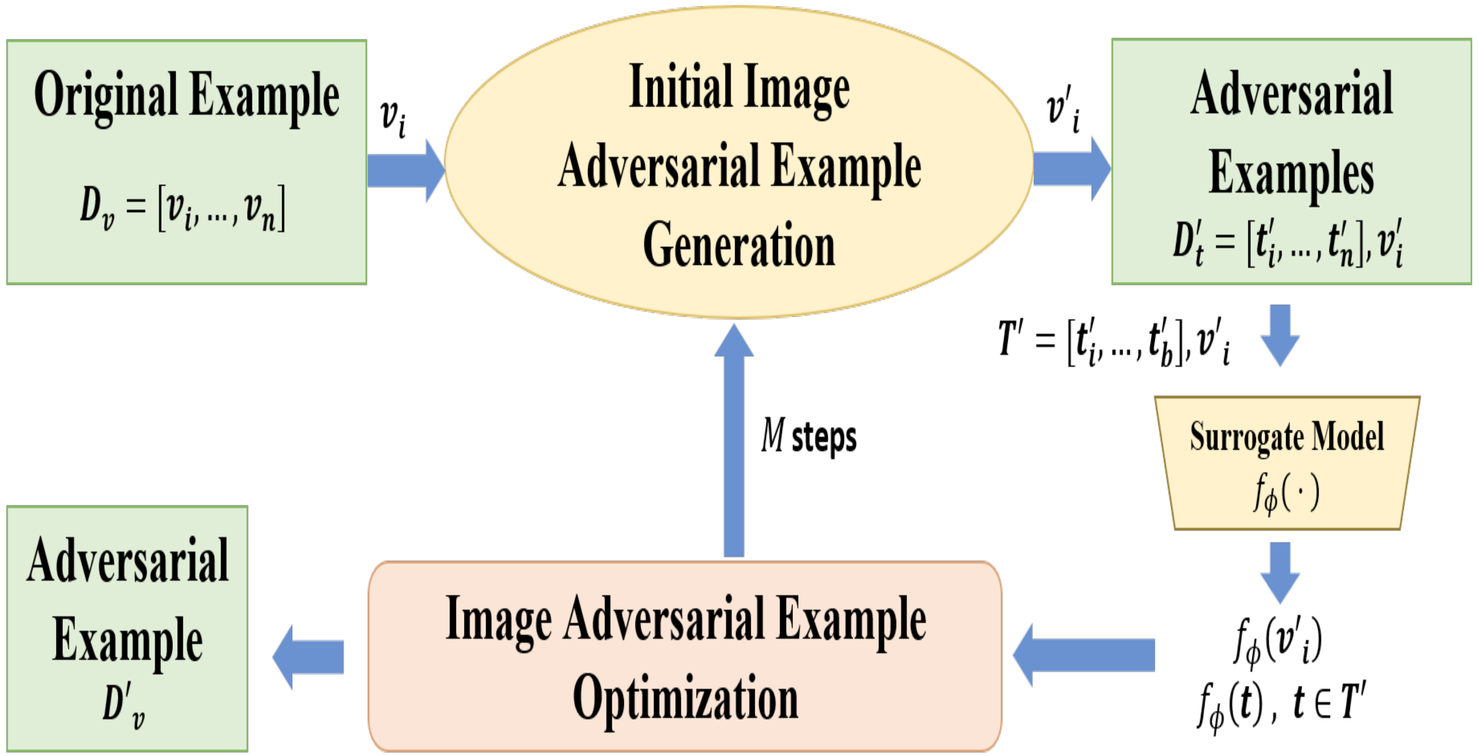
2 Generating Textual Adversarial Example

We generate a set of candidate texts containing all possible single-word synonym replacements and select the text with the lowest cosine similarity to the original image's features as the final adversarial sample.

$$t_i' = \underset{t_i^* \in \mathrm{C}(t_i)}{\operatorname{argmax}} -cos\left(E_T\left(t_i^*, f_\phi\right), E_I\left(v_i, f_\phi\right)\right)$$

HQA-VLAttack

➢ Image Attack



1. Layer-Importance Based Initial Image Adversarial Example Generation

To address the issue of existing methods incorrectly assuming equal layer contributions, we determine layer importance weights by calculating the cosine similarity between the [CLS] token embedding of each layer and that of the top layer. Subsequently, we generate an initial adversarial example that significantly differs from the original image in important layers by minimizing the sum of feature similarities weighted by these importance scores.

2. Contrastive Learning Based Image Adversarial Example Optimization

To further improve the attack success rate, we adopt a contrastive learning approach to optimize the adversarial image. By designing a specific loss function and optimizing it with PGD, we aim to push positive (matching) image-text pairs further apart and pull negative (unmatching) pairs closer in the feature space, thereby inducing the model to retrieve incorrect texts for the adversarial image.

HQA-VLAttack

➢ Image Attack

1 Layer-Importance Based Initial Image Adversarial Example Generation

To address the issue of existing methods incorrectly assuming equal layer contributions, we determine layer importance weights by calculating the cosine similarity between the [CLS] token embedding of each layer and that of the top layer. Subsequently, we generate an initial adversarial example that significantly differs from the original image in important layers by minimizing the sum of feature similarities weighted by these importance scores.

$$w_{i,l} = \cos(E_I(v_i, f_\phi)_{l,1}, E_I(v_i, f_\phi)_{L_p,1})$$

$$\mathcal{L}_l = \sum_{l=1}^{L_p} w_{i,l} \times \frac{1}{D_p} \sum_{j=1}^{D_p} \cos(E_I(v_i, f_\phi)_{l,j}, E_I(v_i', f_\phi)_{l,j})$$

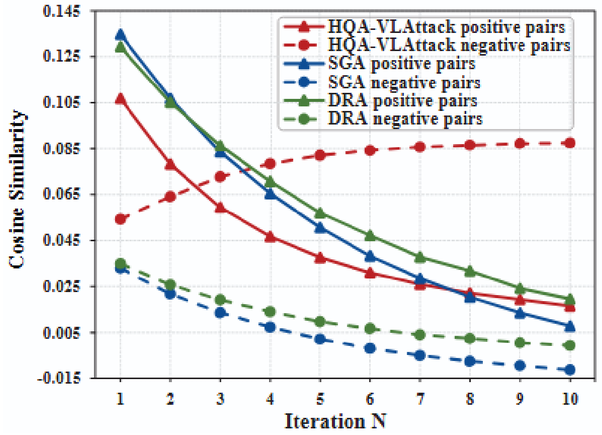
HQA-VLAttack

➢ Image Attack

1 Layer-Importance Based Initial Image Adversarial Example Generation

2 Contrastive Learning Based Image Adversarial Example Optimization

To further improve the attack success rate, we adopt a contrastive learning approach to optimize the adversarial image. By designing a specific loss function and optimizing it with PGD, we aim to push positive (matching) image-text pairs further apart and pull negative (unmatching) pairs closer in the feature space, thereby inducing the model to retrieve incorrect texts for the adversarial image.



$$\mathcal{L}_c = \sum_{v_i^* \in \text{Trans}(v_i')} (\lambda \sum_{t_i' \in T_p} \cos(E_T(t_i', f_\phi), E_I(v_i^*, f_\phi)) + \sum_{t_j' \in T_n} \cos(E_T(t_j', f_\phi), E_I(v_i^*, f_\phi))$$

Experiments

➢ Datasets

- Flickr30K, MSCOCO, RefCOCO+

➢ Baselines

- PGD
- BERT-Attack
- SGA
- Co-Attack
- DRA

➢ Evaluation Metrics

- Attack Success Rate
- B@4, METEOR, ROUGE, CIDEr, SPICE
- Val, TestA, TestB

Experiments

➢ Image-Text Retrieval Comparison

Surrogate Model	Victim Model	Flickr30K Dataset							
		ALBEF		TCL		$CLIP_{ViT}$		$CLIP_{CNN}$	
	Attack Method	TR R@1	IR R@1	TR R@1	IR R@1	TR R@1	IR R@1	TR R@1	IR R@1
ALBEF	PGD	52.45*	58.65*	3.06	6.79	8.96	13.21	10.34	14.65
	BERT-Attack	11.57*	27.46*	12.64	28.07	29.33	43.17	32.69	46.11
	Sep-Attack	65.69*	73.95*	17.60	32.95	31.17	45.23	32.83	45.49
	Co-Attack	77.16*	83.86*	15.21	29.49	23.60	36.48	25.12	38.89
	SGA	97.24*	97.28*	45.42	55.25	33.38	44.16	34.93	46.57
	DRA	96.14*	96.63*	49.74	58.83	39.14	48.39	41.38	51.66
	HQA-VLAttack	**99.79***	**99.98***	**73.02**	**77.60**	**52.15**	**62.05**	**59.64**	**65.59**
TCL	PGD	6.15	10.78	77.87*	79.48*	7.48	13.72	10.34	15.33
	BERT-Attack	11.89	26.82	14.54*	29.17*	29.69	44.49	33.46	46.06
	Sep-Attack	20.13	36.48	84.72*	86.07*	31.29	44.65	33.33	45.80
	Co-Attack	23.15	40.04	77.94*	85.59*	27.85	41.19	30.74	44.11
	SGA	48.91	60.34	98.37*	98.81*	33.87	44.88	37.74	48.30
	DRA	51.09	61.79	98.21*	98.33*	40.25	48.94	42.91	52.49
	HQA-VLAttack	**62.88**	**71.70**	**99.79***	**99.93***	**52.39**	**59.41**	**55.43**	**62.44**
$CLIP_{ViT}$	PGD	2.50	4.93	4.85	8.17	70.92*	78.61*	5.36	8.44
	BERT-Attack	9.59	22.64	11.80	25.07	28.34*	39.08*	30.40	37.43
	Sep-Attack	9.59	23.25	11.38	25.60	79.75*	86.79*	30.78	39.76
	Co-Attack	10.57	24.33	11.94	26.69	93.25*	95.86*	32.52	41.82
	SGA	13.40	27.22	16.23	30.76	99.08*	98.94*	38.76	47.79
	DRA	12.51	30.00	14.65	30.62	98.77*	99.00*	45.47	50.74
	HQA-VLAttack	**25.13**	**41.98**	**24.66**	**44.00**	**100.00***	**100.00***	**74.07**	**77.19**
$CLIP_{CNN}$	PGD	2.09	4.82	4.00	7.81	1.10	6.60	86.46*	92.25*
	BERT-Attack	8.86	23.27	12.33	25.48	27.12	37.44	30.40*	40.10*
	Sep-Attack	8.55	23.41	12.64	26.12	28.34	39.43	91.44*	95.44*
	Co-Attack	8.79	23.74	13.10	26.07	28.79	40.03	94.76*	96.89*
	SGA	11.42	24.80	14.91	28.82	31.24	42.12	99.24*	99.49*
	DRA	12.20	26.59	14.33	29.29	35.21	45.94	99.11*	99.49*
	HQA-VLAttack	**20.75**	**38.66**	**22.13**	**42.45**	**62.82**	**69.46**	**99.87***	**100.00***

➢ Cross-Task ASR Comparison

Attack	ITR → VG			ITR → IC				
	Val ↓	TestA ↓	TestB ↓	B@4 ↓	METEOR ↓	ROUGE-L ↓	CIDEr ↓	SPICE ↓
Baseline	58.46	65.89	46.25	39.7	31.0	60.0	133.3	23.8
Co-Attack	54.26	61.80	43.81	37.4	29.8	58.4	125.5	22.8
SGA	53.55	61.19	43.71	34.8	28.4	56.3	116.0	21.4
DRA	53.88	61.18	43.38	34.8	28.4	56.4	115.9	21.4
HQA-VLAttack	**46.48**	**54.31**	**36.90**	**31.8**	**26.8**	**54.1**	**104.6**	**19.8**

For more experiments, please refer to the original paper.

Conclusion

- We propose a novel black-box adversarial attack method **HQA-VLAttack**.
- By using HQA-VLAttack to generate adversarial examples, we can **generate high quality adversarial examples.**
- Experimental results have shown that HQA-VLAttack is more **practical** and **efficient**.

Thanks for listening!