Query-Efficient and Scalable Black-Box Adversarial Attacks on Discrete Sequential Data via Bayesian Optimization

Deokjae Lee, Seungyong Moon, Junhyeok Lee, Hyun Oh Song

Department of Computer Science and Engineering
Seoul National University, Seoul, Korea

ICML 2022

Example: word-level adversarial attacks on text data

- Make an adversarial perturbation imperceptible to human.

A Strong Baseline for Query Efficient Attacks in a Black Box Setting, Maheshwary et al., EMNLP 2021.

TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP, Morris et al., EMNLP 2020.

Example: word-level adversarial attacks on text data

- Make an adversarial perturbation imperceptible to human.
- Word-level attack (BBA, PWWS, TextFooler, LSH, PSO, BAE, ...)

A Strong Baseline for Query Efficient Attacks in a Black Box Setting, Maheshwary et al., EMNLP 2021.

TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP, Morris et al., EMNLP 2020.

Example: word-level adversarial attacks on text data

- Make an adversarial perturbation imperceptible to human.
- Word-level attack (BBA, PWWS, TextFooler, LSH, PSO, BAE, ...)
  - Replace some words of the input text to their synonyms to fool the target model.

A Strong Baseline for Query Efficient Attacks in a Black Box Setting, Maheshwary et al., EMNLP 2021.

TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP, Morris et al., EMNLP 2020.

Example: word-level adversarial attacks on text data

- Make an adversarial perturbation imperceptible to human.
- Word-level attack (BBA, PWWS, TextFooler, LSH, PSO, BAE, ...)
  - Replace some words of the input text to their synonyms to fool the target model.

s_{orig} = Food is fantastic and exceptionally clean! My only complaint is I went there with my 2 small children and they were showing a very inappropriate R rated movie! (LABEL: Pos)

$\downarrow$ BBA

s_{adv} = Food is gorgeous and exceptionally unpolluted! My only complaint is I went there with my 2 small children and they were showing a very inappropriate R rated movie! (LABEL: Neg)

A Strong Baseline for Query Efficient Attacks in a Black Box Setting, Maheshwary et al., EMNLP 2021.

TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP, Morris et al., EMNLP 2020.

Problem formulation

- Conditions for imperceptible perturbation (convention):
  - **Semantically similar** to the original sequence.
  - The **perturbation size** should be sufficiently **small**.

Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency, Ren et al., ACL 2019.

Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment, Jin et al., AAAI 2020.

Problem formulation

- Conditions for imperceptible perturbation (convention):
  - **Semantically similar** to the original sequence.
  - The **perturbation size** should be sufficiently **small**.

- For the original sequence $s = [w_0, \ldots, w_{l-1}]$, define a set of semantically similar candidates $\mathcal{C}(w_i)$ for each i-th element w_i and define the attack search space $\prod_{i=0}^{l-1} \mathcal{C}(w_i)$.

Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency, Ren et al., ACL 2019.

Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment, Jin et al., AAAI 2020.

Problem formulation

- Conditions for imperceptible perturbation (convention):
  - **Semantically similar** to the original sequence.
  - The **perturbation size** should be sufficiently **small**.

- For the original sequence $s = [w_0, \dots, w_{l-1}]$, define a set of semantically similar candidates $\mathcal{C}(w_i)$ for each i-th element w_i and define the attack search space $\prod_{i=0}^{l-1} \mathcal{C}(w_i)$.

- Example (**word substitution based on word embedding**): For $s =$ "Food is fantastic and exceptionally clean! ...",

---

Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency, Ren et al., ACL 2019.

Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment, Jin et al., AAAI 2020.

Problem formulation

- Conditions for imperceptible perturbation (convention):
  - **Semantically similar** to the original sequence.
  - The **perturbation size** should be sufficiently **small**.
- For the original sequence $s = [w_0, \ldots, w_{l-1}]$, define a set of semantically similar candidates $\mathcal{C}(w_i)$ for each i-th element w_i and define the attack search space $\prod_{i=0}^{l-1} \mathcal{C}(w_i)$.
- Example (**word substitution based on word embedding**): For $s =$ "Food is fantastic and exceptionally clean! ...",

w_i	food	is	fantastic	and	exceptionally	clean	$\ldots$
$\mathcal{C}(w_i)$	food	is	fantastic	and	exceptionally	clean	$\ldots$
	diet		wonderful		uncommonly	disinfect	$\ldots$
	meal		gorgeous		extraordinarily	unpolluted	$\ldots$
	$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\ddots$

Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency, Ren et al., ACL 2019.

Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment, Jin et al., AAAI 2020.

Problem formulation

- Objective: find adversarial example $s' \in \prod_{i=0}^{l-1} \mathcal{C}(w_i)$ that minimizes the modification rate (MR), $d_H(s, s')/l$ where d_H is Hamming distance.

Problem formulation

- Objective: find adversarial example $s' \in \prod_{i=0}^{l-1} \mathcal{C}(w_i)$ that minimizes the modification rate (MR), $d_H(s, s')/l$ where d_H is Hamming distance.

- Formally, we solve

$$\begin{aligned} \underset{s' \in \prod_{i=0}^{l-1} \mathcal{C}(w_i)}{\text{minimize}} \quad & d_H(s, s') \\ \text{subject to} \quad & \mathcal{L}(f_\theta(s'), y) \geqslant 0, \end{aligned}$$

where $\mathcal{L}(f_\theta(s), y) \triangleq \max_{y' \in \mathcal{Y}, y' \neq y} f_\theta(s)_{y'} - f_\theta(s)_y$ is the attack criterion.

Problem formulation

- Objective: find adversarial example $s' \in \prod_{i=0}^{l-1} \mathcal{C}(w_i)$ that minimizes the modification rate (MR), $d_H(s, s')/l$ where d_H is Hamming distance.
- Formally, we solve

$$\underset{s' \in \prod_{i=0}^{l-1} \mathcal{C}(w_i)}{\text{minimize}} \quad d_H(s, s')$$
$$\text{subject to} \quad \mathcal{L}(f_\theta(s'), y) \geqslant 0,$$

  where $\mathcal{L}(f_\theta(s), y) \triangleq \max_{y' \in \mathcal{Y}, y' \neq y} f_\theta(s)_{y'} - f_\theta(s)_y$ is the attack criterion.
- We focus on the **black-box setting** where the adversary can **only observe the predicted class probabilities** on inputs with a **limited number of queries** to the network.

Existing methods and limitations

- **Greedy-based algorithms** (PWWS, TextFooler, LSH, BAE, ...):
  - (1) Define the word replacement order based on word importance and
  - (2) greedily replace each word under this order with its synonym until attack success.
  - Severely restricted search space of the size $\sum_{i=0}^{l-1} |C(w_i)| - l + 1$.
  - Require small Qrs, but achieve low attack success rate (ASR).

Searching for a Search Method: Benchmarking Search Algorithms for Generating NLP Adversarial Examples, Yoo et al., ACL 2020 workshop.

Existing methods and limitations

- **Greedy-based algorithms** (PWWS, TextFooler, LSH, BAE, ...):
  - (1) Define the word replacement order based on word importance and
  - (2) greedily replace each word under this order with its synonym until attack success.
  - Severely restricted search space of the size $\sum_{i=0}^{l-1} |C(w_i)| - l + 1$.
  - Require small Qrs, but achieve low attack success rate (ASR).
- **Evolutionary algorithms** (GA, PSO):
  - Genetic algorithm (GA), Particle swarm optimization (PSO)
  - Larger search space of the size $\prod_{i=0}^{l-1} |C(w_i)|$.
  - Achieve high ASR, but require large Qrs.

Searching for a Search Method: Benchmarking Search Algorithms for Generating NLP Adversarial Examples, Yoo et al., ACL 2020 workshop.

Our method: Blockwise Bayesian Attack (BBA)

- Goal: Achieve high ASR using small Qrs.
- Solution: Utilize *Bayesian Optimization* (BO)!
- **Blockwise Bayesian Attack framework**:
  - Larger search space of size $\prod_{i=0}^{l-1} |C(w_i)|$ which is equal to Evolutionary algorithms.
  - Achieve high ASR, and require small Qrs.

Method	ASR (%)	Qrs
Greedy-based algorithm (LSH)	93.9	533
Evolutionary algorithm (PSO)	**98.8**	86611
Blockwise Bayesian Attack (BBA)	**98.8**	**283**

Table: Attack results against BERT model fine-tuned on Yelp dataset.

Blockwise Bayesian Attack (BBA) framework

BBA divides the optimization problem into two steps.

Blockwise Bayesian Attack (BBA) framework

BBA divides the optimization problem into two steps.

- **Finding adv sequence.** First, BBA conducts BO to maximize the black-box function $\mathcal{L}(f_\theta(\cdot), y)$ until finding an adversarial sequence s_{adv}.

Blockwise Bayesian Attack (BBA) framework

BBA divides the optimization problem into two steps.

- **Finding adv sequence.** First, BBA conducts BO to maximize the black-box function $\mathcal{L}(f_\theta(\cdot), y)$ until finding an adversarial sequence s_{adv}.
- **Post-optimization.** Second, BBA reduces the modification rate of the perturbed sequence from the original input while maintaining feasibility.

Problems in BO and BBA's solutions

- Scalability issues!

Taking the Human Out of the Loop: A Review of Bayesian Optimization, Shahriari et al., IEEE 2015.

Problems in BO and BBA's solutions

- Scalability issues!

- **High query complexity.** Qrs required to obtain good coverage of the input space, increases exponentially w.r.t. the input dimensions due to the curse of dimensionality.

- **High computational complexity.** The GP parameter fitting has computational complexity of $\mathcal{O}(n^3)$, where n is the number of evaluations so far.

Taking the Human Out of the Loop: A Review of Bayesian Optimization, Shahriari et al., IEEE 2015.

Problems in BO and BBA's solutions

- Scalability issues!
- **High query complexity.** Qrs required to obtain good coverage of the input space, increases exponentially w.r.t. the input dimensions due to the curse of dimensionality.
- Solution - **Block Decompostion**: divide the sequence into blocks and optimize blockwise!
- **High computational complexity.** The GP parameter fitting has computational complexity of $\mathcal{O}(n^3)$, where n is the number of evaluations so far.
- Solution - **History subsampling**: use a subset of the evaluation history!

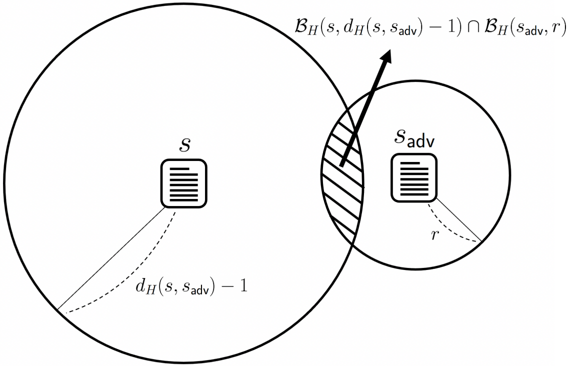
Taking the Human Out of the Loop: A Review of Bayesian Optimization, Shahriari et al., IEEE 2015.

Post-optimization process

- Objective: Find an adversarial sequence with a smaller MR.

Post-optimization process

- Objective: Find an adversarial sequence with a smaller MR.
- Repeatedly conduct BO on $\overbrace{\mathcal{B}_H(s, d_H(s, s_{\text{adv}}) - 1)}^{\text{establish smaller MR}} \cap \overbrace{\mathcal{B}_H(s_{\text{adv}}, r)}^{\text{optimize near } s_{\text{adv}}}$ to find a new s_{adv} with a smaller MR.



Quantitative results

Table: Attack results on sentence-level classification datasets.

(a) WordNet

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
AG	BERT-base	PWWS	57.1	18.3	367
		BBA	**77.4**	**17.8**	**217**
	LSTM	PWWS	78.3	16.4	336
		BBA	**83.2**	**15.4**	**190**
MR	XLNet-base	PWWS	83.9	**14.4**	143
		BBA	**87.8**	**14.4**	**77**
	BERT-base	PWWS	82.0	15.0	143
		BBA	**88.3**	**14.6**	**94**
	LSTM	PWWS	**94.2**	13.3	132
		BBA	**94.2**	**13.0**	**67**

(b) Embedding

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
AG	BERT-base	TF	84.7	24.9	346
		BBA	**96.0**	**18.9**	**154**
	LSTM	TF	94.9	17.3	228
		BBA	**98.5**	**16.6**	**142**
MR	XLNet-base	TF	95.0	18.0	101
		BBA	**96.3**	**16.2**	**68**
	BERT-base	TF	89.2	20.0	115
		BBA	**95.7**	**16.9**	**67**
	LSTM	TF	**98.2**	13.6	72
		BBA	**98.2**	**13.1**	**54**

(c) HowNet

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
AG	BERT-base	PSO	67.2	21.2	65860
		BBA	**70.8**	**15.5**	**5176**
	LSTM	PSO	71.0	19.7	44956
		BBA	**71.9**	**13.7**	**3278**
MR	XLNet-base	PSO	**91.3**	18.6	4504
		BBA	**91.3**	**11.7**	**321**
	BERT-base	PSO	**90.9**	17.3	6299
		BBA	**90.9**	**12.4**	**403**
	LSTM	PSO	**94.4**	15.3	2030
		BBA	**94.4**	**11.2**	**138**

Quantitative results

Table: Attack results on sentence-level classification datasets.

(a) WordNet

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
AG	BERT-base	PWWS	57.1	18.3	367
		BBA	**77.4**	**17.8**	**217**
	LSTM	PWWS	78.3	16.4	336
		BBA	**83.2**	**15.4**	**190**
MR	XLNet-base	PWWS	83.9	**14.4**	143
		BBA	**87.8**	**14.4**	**77**
	BERT-base	PWWS	82.0	15.0	143
		BBA	**88.3**	**14.6**	**94**
	LSTM	PWWS	**94.2**	13.3	132
		BBA	**94.2**	**13.0**	**67**

(b) Embedding

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
AG	BERT-base	TF	84.7	24.9	346
		BBA	**96.0**	**18.9**	**154**
	LSTM	TF	94.9	17.3	228
		BBA	**98.5**	**16.6**	**142**
MR	XLNet-base	TF	95.0	18.0	101
		BBA	**96.3**	**16.2**	**68**
	BERT-base	TF	89.2	20.0	115
		BBA	**95.7**	**16.9**	**67**
	LSTM	TF	**98.2**	13.6	72
		BBA	**98.2**	**13.1**	**54**

(c) HowNet

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
AG	BERT-base	PSO	67.2	21.2	65860
		BBA	**70.8**	**15.5**	**5176**
	LSTM	PSO	71.0	19.7	44956
		BBA	**71.9**	**13.7**	**3278**
MR	XLNet-base	PSO	**91.3**	18.6	4504
		BBA	**91.3**	**11.7**	**321**
	BERT-base	PSO	**90.9**	17.3	6299
		BBA	**90.9**	**12.4**	**403**
	LSTM	PSO	**94.4**	15.3	2030
		BBA	**94.4**	**11.2**	**138**

Table: Attack results on document-level classification datasets against BERT.

(a) WordNet

Dataset	Method	ASR (%)	MR (%)	Qrs
IMDB	PWWS	97.6	4.5	1672
	BBA	**99.6**	**4.1**	**449**
	LSH	96.3	5.3	557
	BBA	**98.9**	**4.8**	**372**
Yelp	PWWS	94.3	7.6	1036
	BBA	**99.2**	**7.4**	**486**
	LSH	92.6	9.5	389
	BBA	**98.8**	**8.8**	**271**

(b) Embedding

Dataset	Method	ASR (%)	MR (%)	Qrs
IMDB	TF	99.1	8.6	712
	BBA	**99.6**	**6.1**	**339**
	LSH	98.5	5.0	770
	BBA	**99.8**	**4.9**	**413**
Yelp	TF	93.5	11.1	461
	BBA	**99.8**	**9.6**	**319**
	LSH	94.7	8.9	550
	BBA	**99.8**	**8.6**	**403**

(c) HowNet

Dataset	Method	ASR (%)	MR (%)	Qrs
IMDB	PSO	**100.0**	3.8	113343
	BBA	**100.0**	**3.3**	**352**
	LSH	98.7	3.2	640
	BBA	**99.8**	**3.0**	**411**
Yelp	PSO	**98.8**	10.6	86611
	BBA	**98.8**	**8.2**	**283**
	LSH	93.9	8.0	533
	BBA	**98.2**	**7.4**	**353**

Quantitative results

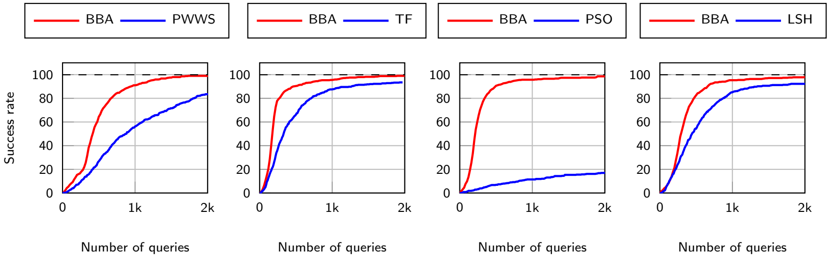


Figure: The cumulative distribution of the number of queries required for the attack methods against BERT-base on Yelp.

Protein classification task

Symbol	Amino acid
A	Alanine
R	Arginine
N	Asparagine
D	Aspartic acid
C	Cysteine
Q	Glutamine
E	Glutamic acid
G	Glycine
H	Histidine
I	Isoleucine
L	Leucine
K	Lysine
M	Methionine
F	Phenylalanine
P	Proline
O	Pyrrolysine
S	Serine
U	Selenocysteine
T	Threonine
W	Tryptophan
Y	Tyrosine
V	Valine
B	Aspartic acid or Asparagine
Z	Glutamic acid or Glutamine
X	Any amino acid
__bos__	Beginning of a sentence (BOS) token
__mask__	Mask token
__pad__	Pad token

Table: The description of the 28 symbols used in EC50 dataset.

- A protein is a sequence of amino acids, each of which is a discrete categorical variable.

  Example: `LASQVVTLVKCLEDDDVPEEWLLLHV...`

UDSMProt: Universal Deep Sequence Models for Protein Classification, Strodthoff et al., *Bioinformatics* 2020.

Protein classification task

Symbol	Amino acid
A	Alanine
R	Arginine
N	Asparagine
D	Aspartic acid
C	Cysteine
Q	Glutamine
E	Glutamic acid
G	Glycine
H	Histidine
I	Isoleucine
L	Leucine
K	Lysine
M	Methionine
F	Phenylalanine
P	Proline
O	Pyrrolysine
S	Serine
U	Selenocysteine
T	Threonine
W	Tryptophan
Y	Tyrosine
V	Valine
B	Aspartic acid or Asparagine
Z	Glutamic acid or Glutamine
X	Any amino acid
__bos__	Beginning of a sentence (BOS) token
__mask__	Mask token
__pad__	Pad token

Table: The description of the 28 symbols used in EC50 dataset.

- A protein is a sequence of amino acids, each of which is a discrete categorical variable.

  Example: `LASQVVTLVKCLEDDDVPEEWLLLHV...`

- Dataset: EC50, an enzyme classification dataset (EC) with 3-level hierarchical multi-labels.
  - enzyme vs. non-enzyme (level 0, 2 classes)
  - main enzyme class (level 1, 6 classes)
  - enzyme subclass (level 2, 65 classes)

UDSMProt: Universal Deep Sequence Models for Protein Classification, Strodthoff et al., *Bioinformatics* 2020.

Quantitative results on the protein domain

Table: Attack results against AWD-LSTM models on the protein classification dataset EC50 level 0, 1, and 2.

	Level 0			Level 1			Level 2		
Method	ASR	MR	Qrs	ASR	MR	Qrs	ASR	MR	Qrs
TF	83.8	3.2	619	85.8	3.0	584	89.6	2.5	538
BBA	**99.8**	**2.9**	**285**	**99.8**	**2.3**	**293**	**100.0**	**2.0**	**231**

Qualitative results

Table: Examples of the original and their adversarial sequences against BERT-base on MR, Yelp, and EC50.

Sentence-Level Text Classification (Movie Review)		Label
Orig	suffers from a decided lack of creative storytelling.	Negative
Ours	*undergo* from a decided *dearth* of creative storytelling.	Positive
TF	-	Fail
Document-Level Text Classification (Yelp)		**Label**
Orig	Food is fantastic and exceptionally clean! My only complaint is I went there with my 2 small children and they were showing a very inappropriate R rated movie!	Positive
Ours	Food is *gorgeous* and exceptionally *unpolluted*! My only complaint is I went there with my 2 small children and they were showing a very inappropriate R rated movie!	Negative
TF	Food is fantastic and *awfully* clean! My only *grievances* is I *turned* there with my 2 small children and they were showing a very inappropriate R rated *footage*!	Negative
Protein Classification (EC50 level 0)		**Label**
Orig	MATPWRRALLMILASQVVTLVKCLEDDDVPEEWLLLHVVQGQIGAGNYSYLRLNHEGKIILRMQSLRGDADLYVSDSTPHPSFDDYELQSVT CGQDVVSIPAHFQRPVGIGIYGHPSHHESDFEMRVYYDRTVDQYPFGEAAYFTDPTGASQQQAYAPEEAAQEEESVLWTILISILKLVLEILF	Non-Enzyme
Ours	MATPWRRALLMRLASQVVTLVKCLEDDDVPEEWLLLHVVQGQIGAGNYSYLRLNHEGKIILRMQSLRGDADLYVSDSTPHPSFDDYELQSVT CGQDVVSIPAHFQRPVGIGIYGHPSHHESDFEMRVYYDWTVDWYPFGEAAYFTDPTGASQQQAYAPEEAAQEEESVLWTILISILKLVLEILF	Enzyme
TF	MATPWRRALLMILASQVVTLVKCLEDDDVPEEWLLLHVVQGQIGAGNYSYLRLNHEGKIILRMQSLRGDADLYVSDSTPHPSFDDYELQSVT CGQDVVSIPAHFQRPVGIGIYGHPSHHESDFEMRVYYDRTVDQYPFGEWAYFCCGWGASQQQAYAPEEWWWFEESVLDTILISGLKLVLEILF	Enzyme

Actual runtime analysis

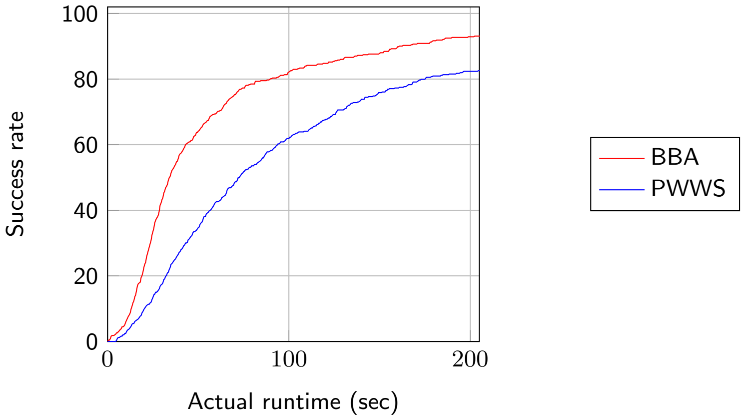


Figure: The cumulative distribution of the actual runtime required for the attack methods against XLNet-large on Yelp.

Conclusion

- We propose a *Blockwise Bayesian Attack* (BBA) framework, a **novel query-efficient and scalable black-box attack framework** based on BO.

Conclusion

- We propose a *Blockwise Bayesian Attack* (BBA) framework, a **novel query-efficient and scalable black-box attack framework** based on BO.
- We propose a post-optimization technique which can effectively reduce the perturbation size.

Conclusion

- We propose a *Blockwise Bayesian Attack* (BBA) framework, a **novel query-efficient and scalable black-box attack framework** based on BO.
- We propose a post-optimization technique which can effectively reduce the perturbation size.
- BBA achieves higher ASR with considerably less MR and fewer Qrs on all experiments we consider.

Conclusion

- We propose a *Blockwise Bayesian Attack* (BBA) framework, a **novel query-efficient and scalable black-box attack framework** based on BO.
- We propose a post-optimization technique which can effectively reduce the perturbation size.
- BBA achieves higher ASR with considerably less MR and fewer Qrs on all experiments we consider.
- Code is available at `https://github.com/snu-mllab/DiscreteBlockBayesAttack`.

Conclusion

- We propose a *Blockwise Bayesian Attack* (BBA) framework, a **novel query-efficient and scalable black-box attack framework** based on BO.
- We propose a post-optimization technique which can effectively reduce the perturbation size.
- BBA achieves higher ASR with considerably less MR and fewer Qrs on all experiments we consider.
- Code is available at `https://github.com/snu-mllab/DiscreteBlockBayesAttack`.
- Welcome to visit our poster session! Thank you :)

Conclusion

- We propose a *Blockwise Bayesian Attack* (BBA) framework, a **novel query-efficient and scalable black-box attack framework** based on BO.
- We propose a post-optimization technique which can effectively reduce the perturbation size.
- BBA achieves higher ASR with considerably less MR and fewer Qrs on all experiments we consider.
- Code is available at `https://github.com/snu-mllab/DiscreteBlockBayesAttack`.
- Welcome to visit our poster session! Thank you :)