Step by Step Loss Goes Very Far: Multi-Step Quantization for Adversarial Text Attacks

Piotr Gaiński
Jagiellonian University
Ardigen
piotr.gainski@doctoral.uj.edu.pl

Klaudia Bałazy
Jagiellonian University
klaudia.balazy@doctoral.uj.edu.pl

Abstract

We propose a novel gradient-based attack against transformer-based language models that searches for an adversarial example in a continuous space of token probabilities. Our algorithm mitigates the gap between adversarial loss for continuous and discrete text representations by performing multi-step quantization in a quantization-compensation loop. Experiments show that our method significantly outperforms other approaches on various natural language processing (NLP) tasks.

1 Introduction

Deep neural networks achieve impressive results, but their vulnerability to adversarial attacks causes major security threats and is a concern when interpreting or explaining model predictions.

In computer vision, the most successful attack methods use gradient-based optimization techniques (Carlini and Wagner, 2017; Madry et al., 2018). They minimize adversarial loss function that encourages the prediction error and imperceptibility of a generated example.

Development of optimization-based attacks in NLP is much more challenging due to the discrete nature of text. Recent methods (Guo et al., 2021; Yuan et al., 2021) overcome this limitation by performing a gradient descent in the continuous space of token representations and eventually quantizing them into discrete text.

A quantization of a token can significantly change its embedding and cause an undesired change of the loss value, degrading the adversarial example. To our knowledge, all existing optimization-based NLP attacks quantize all tokens in a text at once, which creates a considerable gap between adversarial loss for continuous and discrete text representations.

In this paper, we propose MANGO[1] (Multi-step quANtization Gradient-based adversarial Optimizer): a novel optimization-based attack against Transformer (Vaswani et al., 2017) language models that mitigates the aforementioned gap by performing multi-step quantization in a quantization-compensation loop. MANGO quantizes continuous token representations one by one and reoptimizes the adversarial example after each quantization to compensate undesired degradation of adversarial loss value. The construction of MANGO introduces interesting problems that are addressed in Section 3. MANGO achieves superior performance in various NLP tasks, outperforming recent white-box (optimization-based) and black-box attacks.

2 Related Work

Adversarial attacks can be roughly divided into two categories: white-box attacks that have access to the internal model's states (e.g. gradient) and more common black-box attacks that only know outputs of the model. In our paper, we focus on a white-box version of our MANGO attack. In Appendix D, we develop a version of MANGO that can be used in the loosened black-box setting.

Black-Box Methods Most black-box NLP attacks define a space of character or word replacements and heuristically search it for an adversarial example (Yoo et al., 2020). The search space is limited with semantic ad hoc constraints (e.g. limiting edit distance or restricting possible replacements to synonyms) to preserve the attack's imperceptibility. Such constraints disallow some specific perturbations (e.g. replacing a word with its antagonist even if the semantics is preserved in the context of other perturbations) and tend to generate semantically incorrect examples (Morris et al., 2020a).

White-Box Methods Many white-box methods use gradients to guide a heuristic search in a space of text perturbations (Ebrahimi et al., 2018; Cheng et al., 2019; Xu and Du, 2020). Recent methods take a step further and perform gradient descent

[1] Code available at github.com/gmum/MANGO.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2038–2048
May 2-6, 2023 ©2023 Association for Computational Linguistics

optimization. They aim to find an example that minimizes the adversarial loss function, which encourages the prediction error and the imperceptibility of the attack. Because the similarity and fluency of an example are controlled by a powerful external model used in the loss, optimization-based methods do not require hand-crafted semantic constraints, making them more flexible than black-box ones.

Adapting gradient descent in NLP attacks is a challenging problem due to the discrete nature of the optimized text. Yuan et al. (2021) overcome this issue by performing optimization in the continuous space of token embeddings and replacing each token with a possibly new token, which embedding is the closest to the optimized one. An alternative approach is the GBDA method (Guo et al., 2021) that optimizes a continuous distribution of stochastic one-hot vectors and repeatedly samples adversarial examples from the optimized distribution until it fools the attacked model.

Quantization Both methods mentioned above quantize all continuous representations of tokens to a text at once. Quantization of a single token may significantly change its embedding and cause an undesirable change of adversarial loss value. When quantizing all tokens at once, the changes accumulate to a considerable gap between adversarial loss for continuous and discrete text representations (see Section 6). Our MANGO mitigates this gap.

3 MANGO

This section describes our MANGO method. Unlike other optimization-based methods that quantize all token representations at once, MANGO constitutes an entirely new algorithm that quantizes a token and compensates for the resulting change in an adversarial loss value in a step-by-step manner. The construction of MANGO introduces interesting problems that are addressed in the **Optimization**, **Vector Selection** and **Candidates Selection** paragraphs and are further evaluated in Section 5.

Continuous Token Representation The first learnable layer of Transformer takes as input a sequence of tokens $x = (t_1, ..., t_n)$, where $t_i \in 2^{|V|}$ has a single non-zero binary value at index k indicating that it represents the k-th token in vocabulary V.

Similarly to Guo et al. (2021), we relax the input sequence x and replace one-hot encodings t_i with probability vectors π_i. Because the first learnable Transformer layer is a simple linear layer, it can take probability vectors as input without any modification.

A probability vector π_i constitutes probability distribution over tokens from V.

In the embedding layer, the Transformer embeds probability vectors with the function e:

$$e(\pi_i) = \sum_{j=1}^{|V|} (\pi_i)_j E_j, \tag{1}$$

where E_j is the embedding vector of the j-th token. If π_i is quantized, meaning it is a one-hot vector representing some token k, function e simply looks up the k-th embedding: $e(\pi_i) = E_k$. In MANGO, π_i is a probabilistic vector, and its embedding $e(\pi_i)$ is a mixture of embeddings of all tokens weighted by their probabilities π_i. We parameterize π_i with logits Θ_i and a standard softmax function σ, so that $\pi_i = \sigma(\Theta_i)$ and $x = \sigma(\Theta)$ for $\Theta = (\Theta_1, ..., \Theta_n)$.

Loss function Let $m : X \to \mathbb{R}^{|Y|}$ be a classifier that outputs logit vectors and properly predicts a label $y \in Y$ for some datapoint $x \in X$, meaning that $\arg\max_k m(x)_k = y$. An adversarial example is a sample $x' \in X$ that is imperceptible (according to specified criteria) from x but changes the output of the model. In an optimization-based setting, searching for an adversarial example is usually defined as a minimization of an adversarial loss function.

Following Guo et al. (2021), we compose our adversarial loss $\mathcal{L}$ as a combination of margin loss l_m, fluency loss l_f, and similarity loss l_s:

$$\mathcal{L}(x') = l_m(m, x', y) + \lambda_f l_f(g, x') + \lambda_s l_s(g, x', x), \tag{2}$$

where λ_f and λ_s are the coefficients used to balance the losses and g is a reference model.

Margin loss l_m encourages model m to missclassify x' by a margin κ:

$$l_m(m, x', y) = \max(m(x')_y - \max_{k \neq y} m(x')_k + \kappa, 0).$$

Fluency loss l_f promotes x' with a high probability of being generated by a causal language model g that predicts the next token distribution:

$$l_f(g, x') = -\sum_{i=1}^{n} \sum_{j=1}^{|V|} (\pi_i)_j g(\pi_1, ..., \pi_{i-1})_j.$$

Similarity loss l_s is based on BERTScore (Zhang et al., 2020) and captures the semantic similarity between x and x' using contextualized embeddings of tokens $\phi_g(x) = (v_1, ..., v_n)$ and $\phi_g(x') =$

$(v'_1, ..., v'_n)$ produced by the reference model g :

$$l_s(g, x', x) = -\sum_{i=1}^{n} w_i \max_j v_i^T v'_j,$$

where w_i is the inverse frequency of token t_i.

Quantization-Compensation Loop MANGO algorithm searches for a x' that minimizes $\mathcal{L}$, quantizing and compensating it step by step. Algorithm 1 introduces the idea of MANGO.

In the first line, the parameters Θ of x' are initialized, so that $\Theta'_{ij} = C \cdot (x_i)_j$ for some constant C. Each loop starts with **optimization** of x' with respect to $\mathcal{L}$. Then **vector selection** is performed to select π'_i from x' which will be quantized in the current step. Given π'_i, MANGO performs **candidates selection** and selects m the most promising tokens $c_1, ..., c_m$ to which x'_i can be quantized. In the 6th line, each candidate c_j is evaluated by computing $\mathcal{L}$ for a sequence x' with vector π'_i quantized to c_j. Finally, π'_i is quantized to the best c_j chosen from the previous step. Quantized π'_i will no longer be updated during optimization. MANGO repeats lines 2-7 until all vectors in x' are quantized.

Algorithm 1: MANGO

```
  Data: adversarial loss $\mathcal{L}$ (eq. 2)
  Result: sentence $x'$ that minimizes $\mathcal{L}$
1 initialize $x' = (\pi'_1, ..., \pi'_n)$
2 while $x'$ is not fully quantized do
3     optimization: optimize parameters of $x'$
4     vector selection: select probabilistic
        vector $\pi'_i$ from $x'$ for quantization
5     candidates selection: select $m$ tokens
        candidates from $\pi'_i$
6     evaluate these $m$ candidates with loss $\mathcal{L}$
7     quantize $\pi'_i$ to best evaluated token
```

Optimization We optimize x' with the Adam optimizer (Kingma and Ba, 2014) which is reset after each quantization (see Section 5). This allows x' to rapidly change its trajectory to compensate for the degradation of $\mathcal{L}$. The initial number of optimization steps is S, but it decreases by a factor of 2 in each loop to reduce computational costs.

Vector Selection In line 4th, we choose vector π'_i with the highest entropy (see Section 5), because its quantization will introduce the most significant change to x' and is likely to increase the loss value the most. Intuitively, we want such degrading quantizations to occur early in the algorithm, because the more vectors are not quantized yet, the larger capacity x' has to compensate for degradation by finding another local minimum of $\mathcal{L}$.

Candidates Selection In this phase, we select m tokens that can be used to quantize the probability vector π'_i with possibly a small degradation of $\mathcal{L}$. Quantization of π'_i with token k is a step $q_k = (-(\pi'_i)_1, -(\pi'_i)_2, ..., 1 - (\pi'_i)_k, ..., -(\pi'_i)_n)$ in the π'_i space. As π'_i is likely to be in the proximity of its local minimum with respect to $\mathcal{L}$, we want the step q_k to have (1) the lowest norm $\|q_k\|$ possible and (2) follow the direction of the local (minus) gradient. We use this intuition in the formulation of the token score s_k, which is a weighted mean of the probability $(\pi'_i)_k$ and the direction score d_k:

$$s_k = \lambda_{prob}(\pi'_i)_k + (1 - \lambda_{prob})d_k. \tag{3}$$

Note that $(\pi'_i)_k$ is inversely proportional to $\|q_k\|$. We define d_k as cosine similarity between q_k and the local (minus) gradient (see Section 5):

$$d_k = \frac{q_k \left(-\nabla_{\pi'_i}\mathcal{L}(x')\right)^T}{\|q_k\| \cdot \|\nabla_{\pi'_i}\mathcal{L}(x')\|} \tag{4}$$

We then select m tokens with the highest scores s_k.

4 Experiments

In this section, we evaluate MANGO on various NLP tasks and compare it to recent NLP attacks.

Baselines We compare our method with the latest white-box GBDA attack (Guo et al., 2021), as well as recent black-box attacks implemented in TextAttack (Morris et al., 2020b): BERT-Attack (Li et al., 2020), BAE (Garg and Ramakrishnan, 2020) and TextFooler (Jin et al., 2020). To emphasize the importance of multi-step quantization, we evaluate the Naive version of MANGO that performs quantization in one step. MANGO, Naive and GBDA attacks use identical loss. All hyperparameters are listed in appendix A.

Tasks We attack BERT models from TextAttack fine-tuned on three text classification tasks: AG News (Zhang et al., 2015), Yelp Reviews (Zhang et al., 2015), IMDB (Maas et al., 2011), and MNLI task for natural language inference, (Williams et al., 2018). In MNLI p., an attack is allowed to modify only the premise, and in MNLI h., only the hypothesis. For each task, we randomly select 1000 attack targets from the training set. We use a training set as it provides more challenging targets and is more relevant to Adversarial Training (Bai et al., 2021).

Task	Method	Adv.	Adv. prob.	USE sim.	BERTScore	Δ perp.	Δ gram.	# queries
AG News (99.6)	TextFooler	16.2	43.7 ± 26.0	0.81 ± 0.13	0.83 ± 0.10	373 ± 548	0.26 ± 0.69	334 ± 224
	Bert-Attack	20.1	45.7 ± 27.7	0.83 ± 0.11	0.86 ± 0.09	86 ± 133	0.06 ± 0.49	620 ± 472
	BAE	12.6	41.1 ± 24.1	0.78 ± 0.16	0.84 ± 0.11	157 ± 289	0.07 ± 0.53	424 ± 353
	naive	43.7	44.5 ± 43.1	0.82 ± 0.10	0.87 ± 0.06	67 ± 141	0.13 ± 0.62	102 ± 6
	GBDA	12.9	13.7 ± 29.4	0.72 ± 0.13	0.80 ± 0.09	241 ± 382	0.17 ± 0.72	1098 ± 69
	MANGO	**2.7**	3.2 ± 15.3	0.78 ± 0.10	0.83 ± 0.06	30 ± 108	0.10 ± 0.63	496 ± 125
IMDB (98.2)	TextFooler	0.6	34.1 ± 16.9	0.94 ± 0.08	0.93 ± 0.07	108 ± 214	01.03 ± 1.81	761 ± 1 000
	Bert-Attack	0.6	28.0 ± 18.6	0.96 ± 0.07	0.96 ± 0.05	19 ± 38	0.05 ± 0.65	900 ± 922
	BAE	**0.2**	29.3 ± 18.3	0.95 ± 0.08	0.95 ± 0.06	27 ± 59	0.10 ± 0.76	651 ± 665
	naive	30.5	31.1 ± 42.6	0.86 ± 0.09	0.83 ± 0.10	288 ± 346	1.56 ± 2.75	100 ± 13
	GBDA	6.3	7.0 ± 21.3	0.83 ± 0.11	0.79 ± 0.08	294 ± 271	1.44 ± 2.22	1082 ± 146
	MANGO	0.3	0.7 ± 5.7	0.88 ± 0.07	0.83 ± 0.08	59 ± 73	0.99 ± 2.15	1647 ± 746
Yelp (99.9)	TextFooler	4.5	31.7 ± 22.6	0.92 ± 0.10	0.93 ± 0.06	90 ± 192	0.50 ± 01.06	495 ± 526
	Bert-Attack	**1.9**	28.3 ± 19.1	0.93 ± 0.09	0.94 ± 0.06	16 ± 38	0.00 ± 0.55	665 ± 713
	BAE	2.8	30.5 ± 21.1	0.92 ± 0.11	0.93 ± 0.06	29 ± 130	0.06 ± 0.60	501 ± 525
	naive	35.1	35.8 ± 45.4	0.82 ± 0.13	0.84 ± 0.09	25 ± 84	0.75 ± 1.93	102 ± 3
	GBDA	4.5	4.9 ± 18.3	0.79 ± 0.12	0.81 ± 0.06	5 ± 42	0.37 ± 1.59	1101 ± 35
	MANGO	8.5	8.9 ± 27.4	0.82 ± 0.12	0.80 ± 0.07	-30 ± 38	0.34 ± 1.72	1128 ± 718
MNLI premise (94.7)	TextFooler	94.7	-	-	-	-	-	-
	Bert-Attack	3.9	34.3 ± 23.5	0.93 ± 0.08	0.96 ± 0.04	30 ± 58	0.02 ± 0.26	146 ± 148
	BAE	5.0	34.3 ± 23.5	0.92 ± 0.09	0.95 ± 0.04	42 ± 107	0.01 ± 0.26	112 ± 108
	naive	31.6	33.9 ± 24.0	0.91 ± 0.07	0.94 ± 0.04	64 ± 116	-0.01 ± 0.50	97 ± 23
	GBDA	5.9	30.3 ± 21.9	0.80 ± 0.12	0.87 ± 0.07	301 ± 446	0.09 ± 0.67	1044 ± 247
	MANGO	**2.4**	31.6 ± 23.3	0.88 ± 0.08	0.91 ± 0.05	73 ± 123	0.05 ± 0.60	326 ± 125
MNLI hyp. (94.7)	TextFooler	6.5	35.5 ± 24.2	0.94 ± 0.07	0.95 ± 0.04	77 ± 139	0.13 ± 0.39	77 ± 44
	Bert-Attack	2.6	34.3 ± 24.3	1.00 ± 0.01	0.97 ± 0.03	1 ± 0	0.00 ± 0.06	95 ± 62
	BAE	3.5	34.8 ± 24.4	0.95 ± 0.06	0.97 ± 0.03	29 ± 57	0.03 ± 0.25	74 ± 39
	naive	8.4	32.1 ± 22.7	0.89 ± 0.08	0.93 ± 0.04	115 ± 209	0.07 ± 0.36	97 ± 23
	GBDA	0.6	27.4 ± 21.4	0.81 ± 0.12	0.89 ± 0.06	220 ± 454	0.09 ± 0.42	1044 ± 247
	MANGO	**0.3**	30.0 ± 22.4	0.89 ± 0.09	0.93 ± 0.04	85 ± 155	0.06 ± 0.38	258 ± 68

Table 1: Results for black-box and white-box methods. We report: the initial training accuracy of BERT model (under Task); training accuracy under attack (Adv.); probability of ground-truth label prediction under attack (Adv. prob.); similarity between the original and perturbed text computed with USE (Cer et al., 2018) (USE sim.) and with F1 BERTScore (BERTScore); percent change in perplexity computed with GPT-2 (Radford et al., 2019) (Δ perpl.); increase in the number of grammar errors (Δ gram.) obtained with LanguageTool (github.com/jxmorris12/language_tool_python); average number of queries to a victim model (# queries). We omit results for TextFooler on MNLI p., as it has not generated any adversarial example. We also report standard deviation for each result, except adversarial accuracy as it is simply the percent of successful attacks. Our MANGO method achieves superior results on most tasks while maintaining high semantic similarity and grammar fluency. The best results for Adv. are **bold**.

Results Results can be found in Table 1. Our MANGO substantially reduces the training accuracy of the BERT model in all tasks, while maintaining a high level of semantic similarity to the original input. The attacks of MANGO are difficult (low Adv. prob., which indicates that model misclassifies an example by a large margin), fluent (low Δ perp.) and do not flaw the grammatical correctness (low Δ gram.).

In almost all settings, MANGO outperforms other attacks in terms of training accuracy, which we believe to be the fairest metric for comparing optimization-based methods with black-box ones due to inherent design biases (see Appendix B).

MANGO surpasses the recent state-of-the-art optimization-based GBDA attack in terms of most considered metrics: in terms of Adv. acc. and BERTScore on 4/5 tasks and in terms of USE sim., Δ perpl. and Δ gram. on 5/5 tasks.

Moreover, MANGO achieves considerably better results than its Naive version, emphasizing the importance of multi-step quantization.

Qualitative Results We provide qualitative analysis of a few adversarial examples generated by BAE, GBDA, and MANGO in Appendix C.

5 Ablation Study

In this section, we evaluate three solutions from Section 3 that improve the core idea of multi-step quantization:

1. selection of probability vector to quantization by maximal entropy (instead of minimal entropy, which seems more natural choice),
2. scoring token candidates by weighted mean of token probability and gradient direction score (eq. 4),
3. resetting optimizer after every quantization.

Figure 1 compares different MANGO settings. We may observe that selection of probability vector for quantization by maximal entropy ("max entropy") is better than selection by minimal entropy ("min entropy"). Resetting the optimizer after every quantization enhances the performance for both "max entropy" and "min entropy" settings. Finally, we see that MANGO benefits from using both token's probability and gradient direction to score token candidates.

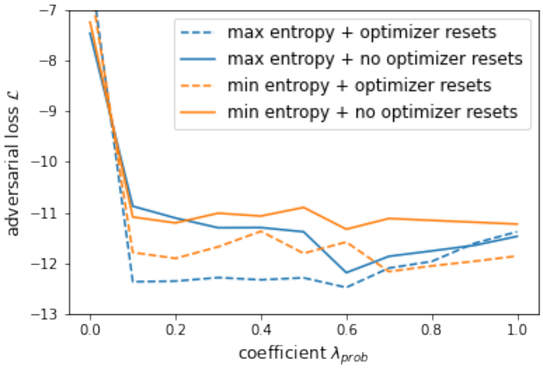

Figure 1: Final adversarial losses for different MANGO setting. "max entropy + optimizer resets" stands for a version of MANGO that selects probability vector for quantization by maximal entropy and resets optimizer after every quantization. Rest of the names follow the same pattern. We also present the influence of the coefficient λ_{prob} used in token candidates scoring function (eq. 4). Loss values are averaged over 10 samples from IMDB dataset.

6 Visualization of Quantization Gap

To visualize the quantization gap between adversarial loss for continuous and discrete text representations, we compared adversarial losses of MANGO, GBDA and a Naive version of MANGO that does not use multi-step quantization. The comparison can be found in Figure 2. We observe that the Naive method converges to the lowest value loss in the optimization phase, but the value explodes after quantization. The GBDA method, which samples probability vectors that resemble discrete one-hot vectors using Gumbel-softmax (Jang et al., 2017), reaches a higher minimum, but its quantization gap is much smaller than that of Naive method. Finally, in the case of MANGO, we observe sudden peaks and slow declines of loss values that correspond to the quantization-compensation loop, in which the quantization of single tokens is followed by the compensation of the quantization gap. After optimization, MANGO continues to quantize tokens step by step further decreasing the loss. MANGO obtains a significantly lower final adversarial loss than GBDA and Naive, avoiding the quantization gap.

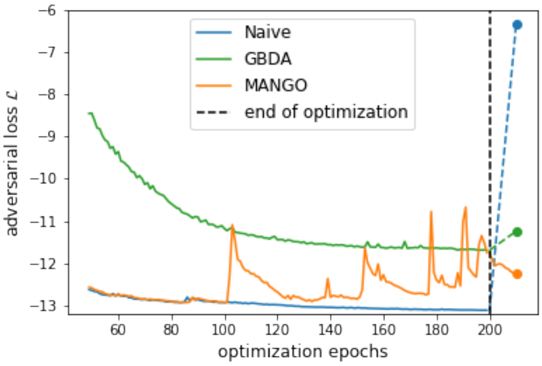

Figure 2: Adversarial loss for epochs 50-200 of optimization for Naive, GBDA and MANGO methods. The vertical dashed line shows the end of optimization. Naive and GBDA methods immediately quantize the tokens, while MANGO do it step by step. The rightmost points shows the final adversarial loss value. We observe that after optimization, MANGO continues to quantize tokens step by step and eventually reaches the best adversarial loss value. Loss values are averaged over 9 samples from IMDB dataset.

7 Conclusion

We developed MANGO, a novel optimization-based attack against Transformer models that mitigates the gap between adversarial loss for continuous and discrete text representations using a quantization-compensation loop. MANGO achieves superior results on various NLP tasks, outperforming recent black-box and optimization-based attacks.

Limitations

One limitation is that the number of queries of MANGO to the attacked model depends on the length of the input sequence. Therefore, MANGO may suffer a long attack time on datasets with long sequences (like IMDB or Yelp).

Moreover, MANGO is restricted only to token replacement. The inability to insert or remove tokens can lead to reduced attack performance.

The most important limitation is the white-box nature of MANGO that excludes it from applications when the internal model's states cannot be known. To partially circumvent this limitation, we propose Gray MANGO - a version of MANGO that can be used in the loosened black-box setting, which we call gray-box setting (see appendix D).

Acknowledgements

The work of Klaudia Bałazy was carried out within the research project "Bio-inspired artificial neural network" (grant no. POIR.04.04.00-00-14DE/18-00) within the Team-Net program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund. Piotr Gaiński and Klaudia Bałazy are affiliated with Doctoral School of Exact and Natural Sciences at the Jagiellonian University.

References

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. 2021. Recent advances in adversarial training for adversarial robustness. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4312–4321. ijcai.org.

Nathaniel Berger, Stefan Riezler, Sebastian Ebert, and Artem Sokolov. 2021. Don't search for a search method - simple heuristics suffice for adversarial text attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8216–8224. Association for Computational Linguistics.

Nicholas Carlini and David A. Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 15–26. ACM.

Xiangyi Chen, Sijia Liu, Kaidi Xu, Xingguo Li, Xue Lin, Mingyi Hong, and David D. Cox. 2019. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7202–7213.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5747–5757. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, and Pramod K. Varshney. 2020. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Process. Mag.*, 37(5):43–54.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 119–126. Association for Computational Linguistics.

Yurii E. Nesterov and Vladimir G. Spokoiny. 2017. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Yining Wang, Simon S. Du, Sivaraman Balakrishnan, and Aarti Singh. 2018. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1356–1365. PMLR.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Jincheng Xu and Qingfeng Du. 2020. Texttricker: Loss-based and gradient-based adversarial attacks on text classification models. *Engineering Applications of Artificial Intelligence*, 92:103641.

Jin Yong Yoo, John X. Morris, Eli Lifland, and Yanjun Qi. 2020. Searching for a search method: Benchmarking search algorithms for generating NLP adversarial examples. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2020, Online, November 2020*, pages 323–332. Association for Computational Linguistics.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between cv and nlp! a gradient-based textual adversarial attack framework.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

A Hyperparameters

MANGO To save computational resources during **candidates selection**, we use the dynamic number of candidates m. We rescale the candidate scores s_k to $[0, 1]$ and take at most $M = 5$ candidates whose scores differ from the best score at most by a threshold $T = 0.5$: $s_k \geq \max_j s_j - T$. We use $\lambda_{prob} = 0.5$ in Equation (4).

White-Box Attacks MANGO, Naive and GBDA methods use the loss function Equation (2) with the same parameters $\lambda_s = 20$, $\lambda_f = 1$, $\kappa = 5$ (taken from Guo et al. (2021)) for all tasks, except Yelp, where they use $\lambda_s = 10$. As a reference model g, we used the GPT-2 model downloaded from the official GBDA repository. We set $C = 10$ for initialization of the adversarial sample parameters. The number of optimization epochs $S = 100$ for all models and the batch size in GBDA was set to 10.

Black-Box Attacks We take TextFooler, BertAttack, and BAE implementations from TextAttack (Morris et al., 2020b) along with their original parameters. For fair comparison, we set the USE similarity threshold to the lowest value (0.2) used along these methods. Following the GBDA paper, we slightly modify the BertAttack method to mitigate its problem with subtokens and extremely long time of attack.

B Comparison Fairness

When comparing the results of optimization-based (MANGO, GBDA, Naive MANGO) and black-box methods (TextFooler, Bert-Attack, BAE), we should note that black-box methods stop perturbing text as soon as they fool the model, while optimization-based attacks minimize adversarial loss (that encourage them to fool the model by some margin) for some fixed number of steps. The former improves similarity metrics (USE sim., BERTScore) and the latter highly decreases the model's prediction on ground-truth labels (Adv. prob.), increasing the difficulty of generated sample. Therefore, we believe that training accuracy under attack (Adv.) is the fairest metric to make a direct comparison between optimization-based and classic black-box methods.

C Attack Examples

To draw some insights into MANGO performance, we compared examples generated by BAE, GBDA and MANGO. We chose all the sentences from AG News and MNLI hypothesis that were successfully perturbed by the three considered methods and on which the methods obtained USE cosine similarity score greater than 0.9. We then sampled two sentences from AG News and two from MNLI hypothesis tasks. To avoid cherry-picking, we fixed a seed and sampled only once. Examples can be found in table 2 and in table 3. We are careful in drawing any conclusion from the qualitative results, however, there seems to be a trend consistent with the result from table 1 and our observations from appendix B: BAE perturbs less words than GBDA and MANGO, but also achieves lower confidence of the mislassified label.

D Gray MANGO

To circumvent the white-box nature of MANGO attack, we additionally develop Gray MANGO: a version of MANGO that can be used in the loosened black-box setting, which we call gray-box setting.

Gray-Box Setting Gray MANGO is not strictly a black-box attack, as it requires the attacked model to take probability vectors and needs access to token vocabulary V. Transformer-based models satisfy these assumptions: they usually share the same V and their embedding function e can be used for both one-hot and probability vectors. However, to avoid misconception, we call this loosened black-box setting a grey-box setting.

Zeroth-Order Optimization Gray MANGO is based on Zeroth-Order Optimization (ZOO) (Nesterov and Spokoiny, 2017). The idea of ZOO is to approximate the gradient using only zeroth order loss values. In computer vision, Chen et al. (2017) developed a ZOO-based attack that significantly outperforms other black-box attacks. We believe that this success can be transferred to the NLP domain. Berger et al. (2021) have proposed an NLP attack that uses a discrete version of ZOO, but the results were unsatisfactory. Our Gray MANGO method is the first to successfully adapt the continuous version of ZOO in NLP attacks.

Formulation The main modification with respect to MANGO is the use of the zeroth-order gradient approximation of the gradient $\nabla_{\Theta'}\mathcal{L}(x')$ (Liu et al.,

Method	Prediction	Sentence
		AG News - Example no 1.
Original	world (100%)	air india trial witness said motivated by revenge (reuters) reuters - a desire for revenge motivated a prosecution witness to tell the air india bombing trial he had been asked to carry an mysterious suitcase on to an airliner, defense lawyers charged on wednesday.
BAE	sci/tech (61%)	air india trial witness said motivated by revenge (reuters) website - a desire for revenge motivated a prosecution witness to tell the air india company s he had been asked to carry an mysterious suitcase on to an account, defense lawyers charged on wednesday.
GBDA	business (99%)	air india trial witness said motivated by revenge - today investigative reuters reporting a desire for revenge motivated criminal prosecution witnesses to tell the air canada strike trial he had been asked to carry an mysterious suitcase on to an airliner, defense lawyers charged on tuesday.
MANGO	business (100%)	air indies trial witness said motivated by revenge (reuters) time - a desire for revenge motivated a prosecution witness to tell the air america arson trial he had been asked to carry a mysterious suitcase on to an airliner, defense lawyers charged on monday.
		AG News - Example no 2.
Original	business (91%)	brazil passes bankruptcy reform brazilian congress gives the green light to a long awaited overhaul of bankruptcy laws, which it hopes will reduce business and credit costs.
BAE	sci/tech (95%)	brazil passes bankruptcy reform brazilian congress gives the green light to a long awaited overhaul of copyright laws, which it hopes will reduce business and credit costs.
GBDA	world (95%)	brazil passes bankruptcy reform brazilian congress gives the green light to a long awaited overhaul of privacy laws, which it aims will reduce tourism and population impacts.
MANGO	world (99%)	brazil passes golf reform brazilian congress gives the green light to a long awaited overhaul of elections laws, which it hopes will reduce spending and maintenance costs.

Table 2: Attack examples sampled from AG News dataset.

2020):

$$\widetilde{\nabla}_{\Theta'}\mathcal{L}(x') = \frac{1}{K}\sum_{i=1}^{K}\frac{\mathcal{L}(\sigma(\Theta' + \mu u_i)) - \mathcal{L}(x')}{\mu}u_i,$$

where u_i is a noise sampled from the normal distribution, μ is the scale factor and $\sigma(\Theta' + \mu u_i)$ is x' with noise μu_i added to its parameters Θ'.

As $\widetilde{\nabla}_{\Theta'}\mathcal{L}(x')$ is unstable, we set $\lambda_{prob} = 1$ and use AMSGrad variant of Adam (Chen et al., 2019) without reset after every quantization. To reduce the high dimensionality of x', which is an issue in ZOO (Wang et al., 2018), we disallow replacement of the original token with tokens that have a cosine similarity of GloVe (Pennington et al., 2014) embedding lower than 0.

Hyperparameters We use almost the same parameters as for MANGO (see appendix A), but with $\lambda_{prob} = 1$, $S = 140$ and $\lambda_s = 80$. To save computational resources, we set $S = 100$ for the IMDB and Yelp datasets. Based on small grid search, we set the noise scaling parameter $\mu = 0.1$.

Method	Prediction	Sentence
MNLI hypothesis - Example no 1.		
Original	contraditcion (96%)	**premise**: the houses are built to a long - standing design and are filled with embroidery, lace, and crochet work. **hypothesis**: there is no embroidery in the houses.
BAE	neutral (45%)	**hypothesis:** there is no fire in the houses.
GBDA	neutral (100%)	**hypothesis:** there is liturgical embroidery in the houses.
MANGO	neutral (99%)	**hypothesis:** there is no erosion in the ruins.
MNLI hypothesis - Example no 2.		
Original	contradiction (100%)	**premise**: whether the service emerges as an adaptation from primary care or as an innovation from the ed is less important than whether it can be evaluated to the satisfaction of those who make key decisions about whether it becomes part of standard practice. **hypothesis:** key decision makers are not important to decided things.
BAE	neutral (96%)	**hypothesis:** consensus decision makers are not important to first things.
GBDA	neutral (98%)	**hypothesis:** key decision makers are noted fairchild – emery associates.
MANGO	neutral (99%)	**hypothesis:** older ahlers are also important in this regard.

Table 3: Attack examples sampled from MNLI hypothesis task.

Results We evaluated the Gray MANGO method and compared it to vanilla MANGO. Results can be found in table 4.

Gray MANGO, which is the first method to incorporate continuous ZOO in NLP attack, performs competitively with other black-box attacks in terms of training accuracy reduction, but struggles to keep adversarial examples similar to original texts. We believe that the performance of Gray MANGO may be greatly elevated by a more thorough design of ZOO components (Liu et al., 2020). This may be an interesting topic for future research.

Task	Method	Adv.	Adv. prob.	USE sim.	BERTScore	Δ perp.	Δ gram.	# queries
AG News (99.6)	TextFooler	16.2	43.7 ± 26.0	0.81 ± 0.13	0.83 ± 0.10	373 ± 548	0.26 ± 0.69	334 ± 224
	Bert-Attack	20.1	45.7 ± 27.7	0.83 ± 0.11	0.86 ± 0.09	86 ± 133	0.06 ± 0.49	620 ± 472
	BAE	12.6	41.1 ± 24.1	0.78 ± 0.16	0.84 ± 0.11	157 ± 289	0.07 ± 0.53	424 ± 353
	G-MANGO	9.7	11.0 ± 25.8	0.57 ± 0.23	0.67 ± 0.14	16k ± 47k	-0.03 ± 0.61	3728 ± 244
	MANGO	**2.7**	3.2 ± 15.3	0.78 ± 0.10	0.83 ± 0.06	30 ± 108	0.10 ± 0.63	496 ± 125
IMDB (98.2)	TextFooler	0.6	34.1 ± 16.9	0.94 ± 0.08	0.93 ± 0.07	108 ± 214	01.03 ± 1.81	761 ± 1 000
	Bert-Attack	0.6	28.0 ± 18.6	0.96 ± 0.07	0.96 ± 0.05	19 ± 38	0.05 ± 0.65	900 ± 922
	BAE	0.2	29.3 ± 18.3	0.95 ± 0.08	0.95 ± 0.06	27 ± 59	0.10 ± 0.76	651 ± 665
	G-MANGO	8.6	10.8 ± 24.3	0.65 ± 0.21	0.66 ± 0.14	16k ± 38k	0.19 ± 1.97	3142 ± 669
	MANGO	0.3	0.7 ± 5.7	0.88 ± 0.07	0.83 ± 0.08	59 ± 73	0.99 ± 2.15	1647 ± 746
Yelp (99.9)	TextFooler	4.5	31.7 ± 22.6	0.92 ± 0.10	0.93 ± 0.06	90 ± 192	0.50 ± 01.06	495 ± 526
	Bert-Attack	**1.9**	28.3 ± 19.1	0.93 ± 0.09	0.94 ± 0.06	16 ± 38	0.00 ± 0.55	665 ± 713
	BAE	2.8	30.5 ± 21.1	0.92 ± 0.11	0.93 ± 0.06	29 ± 130	0.06 ± 0.60	501 ± 525
	G-MANGO	15.7	16.4 ± 32.1	0.62 ± 0.27	0.69 ± 0.15	14k ± 36k	-0.01 ± 1.68	2803 ± 516
	MANGO	8.5	8.9 ± 27.4	0.82 ± 0.12	0.80 ± 0.07	-30 ± 38	0.34 ± 1.72	1128 ± 718
MNLI p. (94.7)	TextFooler	94.7	-	-	-	-	-	-
	Bert-Attack	3.9	34.3 ± 23.5	0.93 ± 0.08	0.96 ± 0.04	30 ± 58	0.02 ± 0.26	146 ± 148
	BAE	5.0	34.3 ± 23.5	0.92 ± 0.09	0.95 ± 0.04	42 ± 107	0.01 ± 0.26	112 ± 108
	G-MANGO	35.1	33.4 ± 23.0	0.77 ± 0.18	0.84 ± 0.10	5876 ± 19k	-0.06 ± 0.64	3158 ± 761
	MANGO	**2.4**	31.6 ± 23.3	0.88 ± 0.08	0.91 ± 0.05	73 ± 123	0.05 ± 0.60	326 ± 125
MNLI h. (94.7)	TextFooler	6.5	35.5 ± 24.2	0.94 ± 0.07	0.95 ± 0.04	77 ± 139	0.13 ± 0.39	77 ± 44
	Bert-Attack	2.6	34.3 ± 24.3	1.00 ± 0.01	0.97 ± 0.03	1 ± 0	0.00 ± 0.06	95 ± 62
	BAE	3.5	34.8 ± 24.4	0.95 ± 0.06	0.97 ± 0.03	29 ± 57	0.03 ± 0.25	74 ± 39
	G-MANGO	9.1	30.8 ± 22.4	0.83 ± 0.13	0.89 ± 0.07	1402 ± 3272	0.04 ± 0.35	3387 ± 807
	MANGO	**0.3**	30.0 ± 22.4	0.89 ± 0.09	0.93 ± 0.04	85 ± 155	0.06 ± 0.38	258 ± 68

Table 4: Comparison of Gray MANGO with black-box methods and vanilla MANGO. We report: the initial training accuracy of BERT model (under Task); training accuracy under attack (Adv.); probability of ground-truth label prediction under attack (Adv. prob.); similarity between the original and perturbed text computed with USE (Cer et al., 2018) (USE sim.) and with F1 BERTScore (BERTScore); percent change in perplexity computed with GPT-2 (Radford et al., 2019) (Δ perpl.); increase in the number of grammar errors (Δ gram.) obtained with LanguageTool (github.com/jxmorris12/language_tool_python); average number of queries to a victim model (# queries). We omit results for TextFooler on MNLI p., as it has not generated any adversarial example. We also report standard deviation for each result, except adversarial accuracy as it is simply the percent of successful attacks. The best results for Adv. are **bold**.