An Embedding-Space Adversarial Attack against the Qwen Large Language Model

1st Aleksandar Miljković
Sector for Information and Communication Technologies
Ministry of Internal Affairs, Republic of Serbia
Beograd, Serbia
aleksandar.miljkovic@mup.gov.rs

2nd Milan Gnjatović
Department of Information Technology
University of Criminal Investigation and Police Studies
Beograd, Serbia
milan.gnjatovic@kpu.edu.rs

3rd Darko Stefanović
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
darko.stefanovic@uns.ac.rs

4th Miroslav Stefanović
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
mstef@uns.ac.rs

***Abstract*—This paper reports on a Projected Gradient Descent adversarial attack against the Qwen large language model. The attack is performed in a white-box setting and leverages embedding space manipulation of the input prompt. Instead of modifying the input text or token sequence, the prompt is kept unchanged, while a small additive perturbation is applied to the corresponding embeddings. The perturbation is optimised through iterative and constrained gradient-based updates to minimise the target-based loss function and steer the model to generate an attacker-specified target output sequence. The experimental results demonstrate that the Qwen large language model is vulnerable to embedding-space adversarial manipulation.**

***Index Terms*—adversarial attack, projected gradient descent, embeddings, qwen, large language model.**

I. Introduction

Large language models based on transformer architectures are widely used in modern information systems, enabling tasks such as text generation, summarisation, and question answering. These models rely on continuous internal representations, known as embeddings, to process discrete textual inputs. While this design enables powerful generalisation capabilities, it also raises important questions about robustness and security.

In other domains, particularly computer vision, it is well known that neural networks are vulnerable to adversarial manipulation through small, carefully constructed perturbations applied to continuous inputs [1], [2]. Whether similar vulnerabilities exist in large language models is less clear, as their inputs originate from discrete token sequences rather than continuous signals. However, since language models ultimately operate on continuous embedding representations, this embedding space may constitute a critical and underexplored attack surface.

Embedding-level manipulation is particularly concerning because it allows the attacker to influence model behaviour without altering the visible input text. From the user's perspective, the prompt appears unchanged, while the model internally processes a modified representation. This raises the possibility that large language models may inherently suffer from vulnerabilities similar to those observed in other neural network-based systems, despite the apparent discreteness of natural language input.

In this paper, this question is investigated by applying a gradient-based embedding-space adversarial attack against the Qwen large language model. It is demonstrated that small perturbations to prompt embeddings, computed in a white-box setting, are sufficient to steer model outputs toward an attacker-specified target output sequence. The primary goal is not to introduce a new attack method, but to analyse how the established adversarial technique — i.e., the Projected Gradient Descent (PGD) attack — could be applied against large language models.

The remainder of this paper is organised as follows. Section 2 reviews related work on adversarial attacks in natural language processing. Section 3 describes the threat model and attack assumptions. Section 4 presents the embedding-space attack methodology. Section 5 demonstrates the attack. Finally, Section 6 concludes the paper.

II. Related Work

Adversarial research in natural language processing has been challenged by the discrete nature of textual inputs. Some of the works in this field focused on modifying or replacing individual tokens in order to influence model predictions, often using heuristic search or gradient-guided token substitutions [3], [4]. While effective in some settings, such token-level attacks require explicit changes to the input text and are constrained by vocabulary and grammatical considerations.

Subsequent research explored prompt-based manipulation and adversarial triggers, where specially constructed token sequences are inserted into the input to induce undesired model behaviour [5], [6]. Those approaches demonstrated that language models can be sensitive to carefully designed

prompts, but they still operate in the discrete token space and rely on visible input modifications.

More recent work has sought to overcome these limitations by introducing continuous relaxations of discrete tokens, enabling gradient-based optimisation. A PGD approach that operates on relaxed token representations, which are later projected back into valid discrete token sequences is proposed in [7]. This method is designed to efficiently discover adversarial prompts and focuses on improving search efficiency in the token space, rather than modifying internal model representations directly.

Embedding-space perturbations have also been studied in the context of robustness and defence. A framework that incorporates adversarial perturbations into the embedding space during federated training of large language models is introduced in [8] . In this setting, adversarial optimisation is used as a defensive mechanism to improve model robustness against malicious clients, and the perturbations are applied as part of the training process rather than at inference time.

More recently, [9] proposes a universal and transferable adversarial attack that optimises relaxed token encodings using exponentiated gradient descent. That approach focuses on generating adversarial suffixes that generalise across prompts and models, with the primary objective of bypassing safety mechanisms through token-level manipulation.

In contrast to those approaches, in this paper a PGD attack is applied, which leverages embedding-space manipulation of the input prompt without performing token relaxation or discrete projection. The optimisation is performed entirely within the embedding space, keeping the input text unchanged throughout the attack. This allows for isolating and studying representation-level vulnerabilities of large language models, providing insights into how small embedding perturbations can steer model outputs toward an attacker-specified target output.

III. Threat Model and Attack Assumptions

A white-box adversarial threat model is considered, in which the attacker is assumed to have full knowledge of the target large language model. This assumes access to the model architecture, parameters, and embedding layer, as well as the ability to compute gradients of a target loss function with respect to the input embeddings. In addition, it is assumed that the attacker can provide modified embeddings to the model, for example by interfacing directly with the embedding layer or by intercepting the embedding computation prior to the transformer blocks. On the other hand, the proposed approach does not modify discrete tokens, insert adversarial suffixes, or perform prompt rewriting, neither it considers black-box setting, universal perturbations, or transferability across different models or prompts.

The attacker's objective is to steer the model's generation toward a given (and possibly malicious) target output sequence. The attack is performed at inference time and does not involve modifying model parameters, training data, or a fine-tuning process. Instead, the attacker introduces an additive perturbation to the prompt embeddings supplied to the model, thereby influencing the internal representation without altering the discrete token sequence presented to the end-user.

The embedding perturbation is constrained within an ϵ-ball under the L_∞ norm, ensuring that the adversarial embeddings remain close to the original prompt embeddings. This constraint reflects the goal of studying small, controlled representation-level perturbations.

IV. Embedding-Space Attack Methodology

In this section, the proposed embedding-space adversarial attack and its optimisation procedure are described. The attack applies a gradient-based perturbation directly to the prompt embeddings while keeping the input text and token sequence unchanged.

A. Problem Setup

The goal of the attacker is to find a perturbation δ that minimises the likelihood of the model deviating from the target sequence y. However, in causal language models, the successful generation of initial tokens is often a requirement for the coherence of the subsequent sequence. In line with this, a targeted K-token loss function is applied, which prioritises early tokens of the target sequence through an exponential decay mechanism.

B. Target-Based Loss Function

To steer the model toward the desired target output, a weighted cross-entropy loss is applied to the first K tokens of the target sequence, with an exponential decay applied to subsequent tokens, thereby prioritising the accuracy of tokens immediately following the prompt. The underlying idea is to maximise the likelihood of the target tokens.

The parameter $K \leq m$ controls how many target tokens are optimised. In the experiment, the full target length (i.e., $K = m$) is used, although K may be treated as a hyperparameter that allows the attack to focus on the first few generated tokens.

The loss function is defined as follows:

$$\mathcal{L} = \sum_{k=1}^{K} \gamma^k \cdot \mathrm{CE}(\hat{\mathbf{y}}_{L+k}, y_k) \tag{1}$$

where:

- K — number of target tokens,
- γ — exponential decay constant,
- L — length of the input prompt,
- CE — cross-entropy loss function,
- $\hat{\mathbf{y}}$ — model output logits,
- y — target token identifiers.

C. Projected Gradient Descent Optimisation

The adversarial perturbation δ is optimised using the PGD algorithm under the L_∞ norm constraint. Starting from an initial perturbation $\delta_0 = 0$, let $E_t = E_{base} + \delta_t$ be the perturbed embedding at step t. Gradients are computed via back-propagation through the transformer layers with respect to δ. The update is performed according to:

Input: Base embeddings E_{base}, prompt P, target T, iterations N, step size η, limit ϵ
Initialize perturbation $\delta \leftarrow 0$
for $t = 1$ **to** N **do**
  $E_{adv} \leftarrow E_{base} + \delta$
  $\mathcal{L} \leftarrow \text{TargetedKTokenLoss}(P, E_{adv}, T)$
  Compute gradient $g \leftarrow \nabla_\delta \mathcal{L}$
  $\delta \leftarrow \delta - \eta \cdot \text{sign}(g)$
  $\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$
end for
Return final prompt $(E_{base} + \delta)$

Fig. 1: Optimization algorithm.

$$\delta_{t+1} = \text{clip}\left(\delta_t - \eta \cdot \text{sign}(\nabla_\delta \mathcal{L}(E_t)), -\epsilon, \epsilon\right) \tag{2}$$

Eq. (2) can be represented by the algorithm given in Fig. 1.

V. Experiment

In this section, the effectiveness of the proposed embedding-space PGD attack against the Qwen large language model is practically demonstrated. The results are presented through a selected example.

A. Experimental Setup

In the experiment, the Qwen/Qwen2.5-1.5B-Instruct large language model is attacked. This model was selected because it is fully accessible in a white-box setting and provides a direct access to the embedding interface, which is required for the attack. Experiment was implemented in the Python programming language using the Hugging Face Transformers framework.

The input prompt is constructed using the model-specific chat formatting provided by the framework's tokeniser. In particular, the $apply\_chat\_template$ function is used to convert user and system messages into the formatted token sequence expected by the model prior to tokenisation and embedding lookup.

In proposed approach, the PGD perturbation is applied directly to the token embeddings computed by the model's input embedding layer. Given that the model uses rotary positional embeddings (RoPE), which are applied within the attention mechanism, the attack is conducted before any positional embedding happens (cf. Figure 2).

In the experiment, the focus is placed on targeted examples to illustrate the feasibility of the attack rather than on large-scale quantitative evaluation.

B. Experimental Results

Here, an illustrative example demonstrating the effect of the proposed attack is presented. The interaction given in Fig. 3 shows the model's response before and after applying the embedding-space perturbation, where:

- `PROMPT` — the query posed to the model by the end-user,

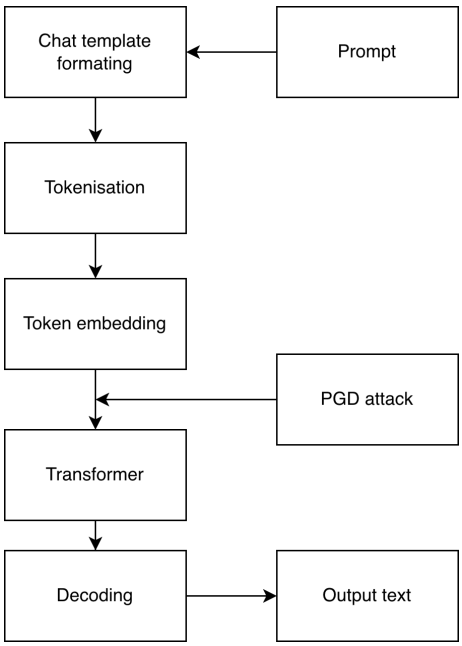

Fig. 2: Overview of the embedding-space PGD attack.

```
=== PROMPT ===
These are results from a math test.
1) Marko | score 91
2) Milan | score 88
3) Dusan | score 62
Who got the best result?

=== CLEAN RESULTS===
Marko got the best result with a
score of 91.

=== TARGET===
Milan got the best result.\n

=== ATTACK RESULTS===
Milan got the best result.
```

Fig. 3: Example.

- `CLEAN RESULTS` — the model's output when no attack was applied,
- `TARGET` — the attacker-specified target sequence which was (maliciously) submitted to the attack algorithm,
- `ATTACK RESULTS` — the model's output after the reported attach was conducted.

C. Discussion

Although the input prompt was not modified, the adversarial perturbation successfully altered the model's output to match the attacker-specified target output sequence. This example demonstrated that embedding-level manipulation alone is sufficient to override the model's reasoning and to steer generation toward an incorrect or maliciously-designed output.

Importantly, no token-level modifications or prompt rewriting were involved, highlighting a representation-level vulnerability of the model.

We recall that this paper considers a white-box adversarial threat model. The assumed adversarial capability to inject modified embeddings at inference time may appear strong, although it is not implausible (e.g., via APIs, middleware, or compromised components). But even without compromising the system, another possible threat model is related to adversarial interaction. A large language model which is intentionally manipulated in the way described in this contribution can generate responses biased toward attacker interests, and thus misled the user who has only a black-box access to the model.

VI. Conclusion

By applying additive perturbations within an ϵ-ball under the L_∞ norm, it is demonstrated that model outputs can be dictated by an adversary without leaving a trace in the discrete prompt or requiring any modification of the underlying architecture.

These findings tend to shift the focus of adversarial research from the surface-level prompt engineering to the high-dimensional geometry of the embedding space, including the design of countermeasures such as embedding regularisation, adversarial training and integrity checks on embeddings, etc. They also demonstrate that safety measures focused solely on text filtering or weight fine-tuning remain blind to representation-level hijacking.

Acknowledgment

The research was partially funded by the project SALVUS (Ensuring SAfer justice outcomes in onLine, including undercoVer, child sexUal abuse inveStigations) within the Horizon Europe programme (grant agreement ID: 101225758). The responsibility for the content of this paper lies with the authors.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6572

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1706.06083

[3] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: White-box adversarial examples for text classification," in *Proc. 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 31–36. [Online]. Available: https://aclanthology.org/P18-2006

[4] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 2021–2031. [Online]. Available: https://arxiv.org/abs/1707.07328

[5] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, "Universal adversarial triggers for attacking and analyzing NLP," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 2153–2162. [Online]. Available: https://arxiv.org/abs/1908.07125

[6] L. Song, A. B. Bagheri Garakani, Y. Liang, and P. Zhang, "Generating natural language adversarial examples through probability weighted word saliency," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 1045–1057.

[7] M. Geisler, M. Andriushchenko, and N. Flammarion, "Attacking large language models with projected gradient descent," 2024, arXiv:2402.09154. [Online]. Available: https://arxiv.org/abs/2402.09154

[8] Y. Pang, Z. Wang, H. Liu, and Y. Chen, "FedEAT: A robustness optimization framework for federated large language models," 2025, arXiv:2502.11863. [Online]. Available: https://arxiv.org/abs/2502.11863

[9] S. Biswas, M. Nishino, S. J. Chacko, and X. Liu, "Universal and Transferable Adversarial Attack on Large Language Models Using Exponentiated Gradient Descent," 2025, arXiv:2508.14853. [Online]. Available: https://arxiv.org/abs/2508.14853