IM-BERT: Enhancing Robustness of BERT through the Implicit Euler Method

Mihyeon Kim[1]* **Juhyoung Park**[2]† **Youngbin Kim**[1, 3]

[1]Department of Artificial Intelligence, Chung-Ang University
[2]School of Computer Science and Engineering, Chung-Ang University
[3]Graduate School of Advanced Imaging Sciences, Multimedia and Film, Chung-Ang University
{mh10967, wngud51, ybkim85}.cau.ac.kr

Abstract

Pre-trained Language Models (PLMs) have achieved remarkable performance on diverse NLP tasks through pre-training and fine-tuning. However, fine-tuning the model with a large number of parameters on limited downstream datasets often leads to vulnerability to adversarial attacks, causing overfitting of the model on standard datasets. To address these issues, we propose IM-BERT from the perspective of a dynamic system by conceptualizing a layer of BERT as a solution of Ordinary Differential Equations (ODEs). Under the situation of initial value perturbation, we analyze the numerical stability of two main numerical ODE solvers: *the explicit and implicit Euler approaches*. Based on these analyses, we introduce a numerically robust IM-connection incorporating BERT's layers. This strategy enhances the robustness of PLMs against adversarial attacks, even in low-resource scenarios, without introducing additional parameters or adversarial training strategies. Experimental results on the adversarial GLUE (AdvGLUE) dataset validate the robustness of IM-BERT under various conditions. Compared to the original BERT, IM-BERT exhibits a performance improvement of approximately 8.3%p on the AdvGLUE dataset. Furthermore, in low-resource scenarios, IM-BERT outperforms BERT by achieving 5.9%p higher accuracy.

1 Introduction

Many Pre-trained Language Models (PLMs) (Kenton and Toutanova, 2019; Min et al., 2021; Raffel et al., 2020) have shown remarkable performances on a wide range of downstream tasks. The high performance of PLMs in domain-specific downstream tasks is achieved through a two-stage training process. In the first stage, PLMs are pre-trained on a high-resource corpus for general representation learning. Subsequently, PLMs are fine-tuned on low-resource data for specific downstream tasks.

On the other hand, this two-stage training process makes PLMs prone to overfitting on the standard dataset, rendering them vulnerable to adversarial attacks through perturbations (Yuan et al., 2021; Jin et al., 2020; Lin et al., 2021; Nie et al., 2020; Ribeiro et al., 2020; Linlin Liu and Bing, 2023). These issues arise from aggressively fine-tuning PLMs with a vast number of parameters on low-resource datasets to enhance performance (Jiang et al., 2020; Brown et al., 2020). Given that PLMs require pre-training on high-resource data to obtain high-quality representations, maintaining the number of parameters is essential. The dilemma suggests addressing the issue during the second step, the fine-tuning stage.

Numerous studies (Aghajanyan et al., 2021; Wu et al., 2022; Jiang et al., 2020) have been conducted in the field of adversarial training, focusing on the fine-tuning stage. They focus on enhancing robustness using various training strategies, including regularization and optimization techniques in backward propagation with dataset-based hyperparameters. Unlike previous approaches, there is a largely unexplored area that enhances robustness against adversarial attacks in forward propagation on standard training. We propose a robust network architecture for PLMs inherently more resistant to adversarial attacks on standard fine-tuning.

From a dynamic system perspective, a plethora of studies (Chen et al., 2018; Lu et al., 2018; Dong et al., 2020) have explored the relationship between neural networks and Ordinary Differential Equations (ODEs), interpreting the residual connection within neural networks as an Euler discretization of ODEs. This insight facilitates the integration of various numerical ODE solvers into neural networks. The numerical solvers can be categorized into two approaches: *explicit and implicit approaches*. Among the two approaches, the im-

*Currently at: KT CORPORATION, mihyeon.gim@kt.com
†Currently at: VAIV COMAPANY, juu9802@vaiv.kr

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 16217–16229
November 12-16, 2024

plicit method itself exhibits high stability against perturbations, such as adversarial attacks (Braun and Golubitsky, 1983). By adopting the implicit method, several studies (Li et al., 2020; Reshniak and Webster, 2020; Yin et al., 2018; Shen et al., 2020) have demonstrated stable performance against both white-box and black-box adversarial attack methods in vision tasks.

Inspired by these previous works, we analyze the theoretical stability of two kinds of numerical ODEs solvers. We consider adversarial attacks as a kind of perturbation on inputs and compare the numerical stability of the two approaches. Based on theoretical analysis, we propose the IM-connection, utilizing the implicit Euler method, to construct a robust architecture against perturbations. By incorporating IM-connection within BERT, we develop IM-BERT to mitigate vulnerabilities to adversarial attacks. This analysis enables us to establish a theoretical basis to construct robust neural architectures without necessitating specialized training strategies or an increased number of parameters.

To validate that our method effectively defends against a wide range of adversarial attacks, we conduct experiments using the Adversarial GLUE (AdvGLUE) dataset (Wang et al., 2021b). Our model outperforms the baseline by up to 4.5%p on average across all tasks in AdvGLUE. Moreover, under stringent conditions with sampled instances, IM-BERT exhibits higher accuracy than BERT, with an improvement of up to 5.9%p on average. To mitigate the time-cost problem, we demonstrate that an IM-BERT model, applying the IM-connection between several layers, is comparable to one with full implicit connections among layers.

Our main contributions are:

- We analyze BERT as an ODE solver and introduce IM-BERT with the IM-connection in an implicit approach. To the best of our knowledge, this is the first work to consider PLMs and enhance their robustness against adversarial attacks from a dynamic system perspective.
- We demonstrate that IM-BERT reinforces resistance to adversarial attacks that perturb at diverse levels. This approach only requires simple modification in standard training.
- We validate its robustness even in low-resource scenarios, suggesting that IM-BERT can maintain stability even when training on scarce datasets.

2 Related Work

2.1 Robust Learning against Adversarial Attack

Numerous studies (Jiang et al., 2020; Zhu et al., 2020; Wang et al., 2021a; Aghajanyan et al., 2021) have devised training strategies aimed at defending against adversarial attacks. One prominent approach is adversarial training, which seeks to produce correct predictions by incorporating adversarial perturbations into the learning process using regularization and optimization techniques.

In terms of regularization, SMART (Jiang et al., 2020) proposes robust fine-tuning using a regularization module and optimization derived from the proximal point method. This approach highlights that overfitting is likely to occur when PLMs are fine-tuned on limited downstream data. Similarly, R3F and R4F (Aghajanyan et al., 2021) fine-tune PLMs rooted in an approximation to trust region, preventing degradation of generalizable representations of PLMs.

To optimize adversarial samples, FreeLB (Zhu et al., 2020) employs a gradient method that perturbs word embeddings to minimize adversarial risks during adversarial training. Like FreeLB, CreAT (Wu et al., 2022) effectively guides the optimization of adversarial samples to deceive the entire model more efficiently.

Training on limited adversarial samples not only makes defending against various attacks challenging but also necessitates the generation of costly adversarial perturbations. Moreover, they do not directly address the intrinsic robustness of the architecture itself. PlugAT (Zheng et al., 2022) has explored the design of inherently robust network architectures by introducing specific modules that enable robust predictions with only a modest increase in the number of trainable parameters. Despite these efforts, the realm of adversarial defense through robust architecture design in forward propagation remains largely unexplored, to our knowledge.

Motivated by the need for a robust architecture that defends against adversarial attacks, we aim to design a resilient architecture for standard fine-tuning. This is achieved even in challenging conditions, by fine-tuning models solely on clean data, focusing on the inherent stability of the architecture.

2.2 ODE Neural Network

The relationship between neural networks and Ordinary Differential Equations (ODEs) has been elucidated from various perspectives (Lu et al., 2018; Kawaguchi, 2020; Lu et al., 2020). These studies demonstrate that the architecture of ResNet (He et al., 2016) and RNN (Schmidt, 2019) can be interpreted as an explicit Euler discretization: the first-order explicit Euler method (Chen et al., 2018; Lu et al., 2018; Dong et al., 2020). Based on this interpretation, robust and efficient models have been proposed, utilizing ODE numerical solvers: explicit and implicit approaches.

With explicit methods, the majority of research has focused on achieving stable architectures by emphasizing higher-order precision or step size. The first-order accuracy of residual connection leads the neural networks to accumulate larger truncation errors (Zhu et al., 2023; Haber and Ruthotto, 2017; Baier-Reinio and De Sterck, 2020; Zhang et al., 2021). One stable architecture (Haber and Ruthotto, 2017) shows stable forward and backward propagations by employing the Leapfrog ODE solver and the Verlet method. Similarly, in NLP, the ODE Transformer (Li et al., 2022) applied the Runge-Kutta method, showcasing its stability by obtaining more precise hidden states and preventing accumulated errors. On the other side, scaling the output of layers by the step size in the residual connection contributes to generating more stable predictions against Gaussian noise (Zhang et al., 2019). By setting the step size to less than 1 or a learnable parameter, the models satisfy the stability region of the explicit method (Yang et al., 2020).

Theoretically, the explicit method has an inherent limitation when the input data is perturbed. The explicit approach may not converge to a solution depending on the step size (Atkinson and Han, 1993). This implies vulnerability to adversarial attacks that deceive the model by perturbing clean data. In contrast, the implicit method exhibits superior numerical stability against adversarial attacks. More robust and stable architectures have been explored by applying the implicit Euler method to residual connections. Previous works have implemented the implicit method in vision by adapting Newton's iteration or gradient descent methods in practice (Shen et al., 2020; Li et al., 2020; Reshniak and Webster, 2020). These studies have demonstrated robustness against noise and adversarial attacks. However, given the non-differentiable and discrete characteristics of text, along with its diverse nature or adversarial attack, a more versatile approach is required to defend against complex attacks (Yuan et al., 2021). Consequently, in extending the research connecting ODEs and neural networks in NLP, we apply the implicit method to BERT, inspired by IE-skips (Li et al., 2020).

3 Method

In this section, we first describe the network as an ODE solver for the initial value problem and compare the numerical stability of the explicit and implicit Euler methods. While the explicit method offers simplicity and speed advantages, it diverges the hidden states when an input is perturbed. Conversely, the implicit method shows absolute stability compared to the explicit method. With this insight, we opt for the implicit method as an ODE solver. Leveraging the gradient descent method to implement the implicit method, we introduce IM-connection and IM-BERT, a more stable approach for PLMs.

3.1 Neural Network as an ODE Solver

Given a dataset with $\{X, Y\} = \{(x, y)|x \in \mathbb{R}^n, y \in C\}$, a neural network maps x to y as a mapping function $\Phi : X \rightarrow Y$. The network consists of n layers, which are nonlinear transformations $\phi_t : H_t \times \Theta_t \rightarrow H_{t+1}$ parameterized by hidden states $h_t \in H$ and parameters $\theta_t \in \Theta_t$ in t-th layer. In general terms, the hidden state h_t is derived from the t-th layer with the input h_{t-1} and θ_t, and is governed by the following equation:

$$h_t = \phi_t(h_{t-1}, \theta_t) \qquad (h_0 = x, t = 1, \ldots, N) \quad (1)$$

Theoretically, as the neural network becomes very deep (i.e., as t approaches ∞), we can formulate the network for continuous t as:

$$h(t) = \phi(x, \theta(t)) \qquad t > 0 \qquad (2)$$

From the perspective of ODEs, we can establish a connection between neural networks and ODEs, where a change of hidden states in the network for t is related to the layer of the network. In the continuous flow of t, the hidden states h vary through the layer. As t flows, the process of changing hidden states $h(t)$ through layers can be formulated into the following ODE:

$$\frac{dh}{dt} = \phi(h(t), \theta(t)) \qquad (3)$$

Given $\phi(x, \theta(t))$, the network seeks all possible hidden states $h(t)$ that enable it to predict the final hidden states y while satisfying Eq. 3 via input x, the initial value of the hidden states. Therefore, a neural network with n layers can be interpreted as an ODE solver to find the solution to the initial value problem:

$$\frac{dh}{dt} = \phi(h(t), \theta(t)) \quad h(0) = x \ , \ t > 0 \quad (4)$$

3.2 Residual Connection as Euler Method

Directly solving the initial value problem of Eq. 4 is generally infeasible. Therefore, it is necessary to use a numerical procedure to approximate a solution. The right side of Eq. 4 is approximated by forward and backward difference methods for the range from t to $t+\gamma$ and from $t-\gamma$ to t, with a step size γ. Applying these methods, both explicit and implicit Euler methods find the hidden states h_t as a solution using layers ϕ_t and the output h_{t-1} of the previous layer, sequentially.

Explicit Method Obtaining h_t is a simple process of updating the previous hidden states h_{t-1} by adding the product of the step size γ and the transformation $\phi_t(h_{t-1}, \theta_t)$ through the t-th layer. The computation h_t is written as:

$$\begin{aligned} h_t &= h_{t-1} + \gamma \ \phi_t(h_{t-1}, \theta_t) \\ h_0 &= x, \ t = 1, ..., N \end{aligned} \quad (5)$$

When the step size γ equals 1, the explicit Euler method serves as the residual connection in the neural network.

Implicit Method To compute h_t in the implicit method, it is essential to solve the following non-linear equation:

$$\begin{aligned} h_t &= h_{t-1} + \gamma \ \phi_t(h_t, \theta_t) \\ h_0 &= x, \ t = 1, ..., N \end{aligned} \quad (6)$$

assuming that $\phi_t(h_t, \theta_t)$ is Lipschitz continuous, this equation has a unique solution if the step size γ is sufficiently small.

Both the explicit and implicit systems generate hidden states inductively, starting with the initial input x. Thus, it is critical to understand whether the two systems can stably approximate the hidden states when perturbations are added to x.

3.3 Stability of Implicit Connection in Perturbation

We first define stability and stable regions when the initial value x is perturbed. We then analyze the stability of each method using the model equation, which is generally used for stability evaluation:

$$\frac{dt}{dh} = \lambda h(t) + \psi(t, x) \quad x = h(0) \ , \ \lambda < 0 \quad (7)$$

Definition 1. (Absolute Stability) The system is absolutely stable for its initial value problem with input x if there exists a step size γ such that the error between hidden states from $x + \eta$ and x converges to zero:

$$\lim_{n \to \infty} (\phi_n(x + \eta, \theta_n) - \phi_n(x, \theta_n)) = 0 \quad (8)$$

Definition 2. (Region of Absolute Stability) The set of all $\lambda\gamma$, in which λ, γ satisfy absolute stability, is called the region of absolute stability.

As shown in the two definitions above, when the initial value is perturbed, the region of absolute stability becomes constrained, as it must satisfy absolute stability. We analyze the region of absolute stability in the implicit and explicit methods in the following propositions:

Proposition 1. (Stability of Explicit Method) For an explicit Euler method, the model equation is absolutely stable if and only if the region of absolute stability is $|1 + \gamma\lambda| < 1$.

Proof 1. Let the solutions of the model equation be h and h_η when the input is x and $x + \eta$ perturbed by η. Subtracting the model equation by the equation whose initial value is perturbed, the error of two solutions can be represented as the initial value problem $h_\eta - h = \frac{d}{dt}(h_\eta - h)$ with $h_\eta(0) - h(0) = \eta$. Applying the explicit method, the error becomes $(h_\eta - h)_n = (1 + \gamma\lambda)^n \eta$ when $t = n$ by inductive argument. This error goes to 0 iff $|1 + \gamma\lambda| < 1$. □

Proposition 2. (Stability of Implicit Method) For an implicit Euler method, the model equation is absolutely stable, regardless of the step size γ.

Proof 2. Let the solutions of the model equation be h and h_η when the input is x and $x + \eta$ perturbed by η. Subtracting the model equation by the equation whose initial value is perturbed, the error of two solutions can be represented as the initial value problem $h_\eta - h = \frac{d}{dt}(h_\eta - h)$ with $h_\eta(0) - h(0) = \eta$. Applying the implicit method, the error becomes $(h_\eta - h)_n = \frac{1}{(1-\gamma\lambda)^n} \eta$ when

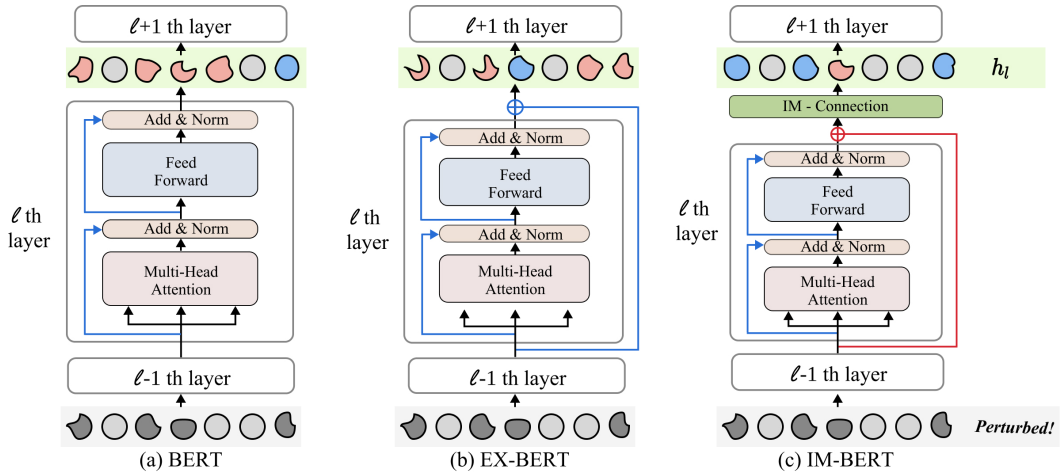

Figure 1: **The l-th layer in each architecture** The perturbed inputs pass through the layer to estimate the hidden states. Blue represents the corrected hidden state, while red indicates the opposite. Blue connections represent explicit residual connections, while red connections represent IM-connections. (a) **BERT** (b) **EX-BERT** In the layer, the layers are connected with the explicit method. (c) **IM-BERT** The hidden states taken from the l-th layer are updated in IM-connection.

$t = n$. The error goes to 0, regardless of the value $\lambda\gamma$. □

Therefore, for the initial value x perturbed by η, the explicit method can diverge in the process of approximating the hidden states h_t in the t-th layer, depending on the step size γ and the stability region. In contrast, the system to which the implicit method is applied exhibits absolute stability with respect to the perturbed x. Proposition 2 indicates that the hidden states are approximated stably, regardless of step size η. From the perspective of adversarial attacks, which degrade the performance of the model by introducing data perturbation η, the system with implicit methods can reliably obtain hidden states. These propositions suggest that this network is more robust against adversarial attacks than the explicit one.

3.4 IM-BERT

We design the network to guarantee more robust hidden states, acting as an ODE solver, especially in situations where it must counter adversarial attacks. As observed with the residual connection, the explicit method has the advantage of a quick and simple forward process. However, according to the above analysis, it may be challenging for the system to remain within the stability region practically, making it potentially vulnerable to adversarial attacks that perturb the data.

$$h_t^* := arg\min_x ||h_t - h_{t-1} - \gamma\phi_t(h_t, \theta_t)||^2 \quad (9)$$

To address this issue, we modify the residual connection by applying an implicit method to ensure the network maintains a stable process. Unlike the residual connection, the implicit method requires an additional process to find the solution to Eq. 6 to obtain the hidden states. Typically, this solution is found using Newton's iteration (Shen et al., 2020; Zhang et al., 2017). However, although this approach may be stable for precise input x, it no longer guarantees absolute stability as it threatens stability when input is perturbed (Atkinson and Han, 1993).

Therefore, inspired by previous works (Li et al., 2020; Reshniak and Webster, 2020), we adopt a gradient descent algorithm to approximate the solution of Eq. 6, i.e., the hidden states. The aim of this algorithm is to find the optimal hidden states that

Algorithm 1 IM connection within IM-BERT layers

Input: Hidden states $h_{t-1}$from previous $(t-1)$-th layer and t-th layer $\phi_t(\cdot)$
Parameter: Iteration number T and step size γ. (We set a step size γ to 0.1 in our experiments.)
Output: Hidden states h_t

Initialize $h_t^0 = h_{t-1} + \phi_t(h_{t-1}, \theta_t)$
for $i = 0, ..., T-1$ **do**
  $loss_i = ||h_t^i - h_{t-1} - \phi_t(h_t^i)||^2$
  Compute gradient $\bigtriangledown_{h_t^i} loss_i$.
  $h_t^i = h_t^i - \gamma \bigtriangledown_{h_t^i} loss_i$
end for
return The hidden states h_t of t-th layer.

		SST-2	QQP	QNLI	MNLI-m	MNLI-mm	RTE	*Avg.*
Adversarial Training	FreeLB	31.6	51	**45.4**	33.5	21.9	42.0	37.6
	CreAT	*35.3*	**51.5**	34.9	36.0	22.0	**45.2**	*39.1*
	R3F	20	38.2	34.9	30.8	33.1	32.2	31.5
	SMART	20.6	39.1	32.3	28.6	32.1	28.6	30.2
Standard Training	BERT	31.1	47.4	36.5	33.1	27.8	34.6	35.1
	EX-BERT	33.8	42.3	40.5	*36.3*	*33.3*	28.4	35.8
	IM-BERT	**39.2**	*48.8*	*41.2*	**38.1**	**35.2**	*44.4*	**41.2**

		SST-2	QQP	QNLI	MNLI-m	MNLI-mm	RTE	*Avg.*
Adversarial Training	SMART$_{large}$	25.21	36.49	34.61	26.89	23.32	38.16	30.29
	SMART$_{RoBERTa}$	50.29	64.22	52.17	*45.56*	36.07	**70.39**	**53.71**
	infoBERT$_{RoBERTa}$	47.61	49.29	54.86	**50.39**	**41.26**	39.47	46.04
	FreeLB$_{RoBERTa}$	**61.69**	42.18	**62.29**	31.59	27.60	*62.17*	50.47
Standard Training	BERT$_{large}$	33.03	37.91	39.77	28.72	27.05	40.46	34.49
	BERT	21.48	41.23	33.47	28.59	28.69	40.79	32.38
	IM-BERT	44.34	36.02	48.86	32.25	*38.89*	43.92	40.71
	IM-RoBERTa	*52.04*	**69.91**	*56.68*	33.68	37.65	60.49	*51.74*

Table 1: **Results on AdvGLUE** (Upper) For dev sets, We report the accuracy, comparing adversarial training and standard training. Adversarial training results are sourced from the CreAT paper. (Bottom) For test sets, We report the accuracy of adversarial training and standard training, with adversarial training results from the AdvGLUE paper.

minimize the transformation represented by Eq. 6.

When initializing h_t in Algorithm 1, the initial value h_t^0 is estimated as an explicit method in terms of ensuring the advantages of forward and backpropagation, like in a previous work (Li et al., 2020). Afterward, IM-connection finds the optimal hidden states h_t^* by calculating the gradient of Eq. 9 in a series of iterations T.

Utilizing the gradient descent method to the implicit approach, We compare the three architectures. Firstly, as shown in Figure 1(a), BERT does not have a residual connection between each layer. BERT goes through the process of monotonically transferring hidden states h_t between layers.

The network can be interpreted as an ODE solver and applied as a layer. Inspired by the analysis, as shown in Figure 1(b) and (c), we propose EX-BERT and IM-BERT that transform the connection between layers into explicit and implicit Euler methods, respectively. We implement EX-BERT by adding residual connections between layers in BERT. It seems similar to IB-BERT (Sun et al., 2020). But for a fair comparison with BERT and IM-BERT, we apply only the residual connection among the layers except for layer normalization (Ba et al., 2016).

We construct IM-BERT by incorporating BERT and the IM-connection using the gradient descent algorithm. In particular, we choose not to transform all residual connections linking feed-forward and multi-head attention. This decision stems from our consideration of distinct properties between different modules and the trade-off between robustness and time cost. Even if robustness is enhanced through fine-tuning with IM-connection, applying IM-connection to all residual connections causes excessive time costs. Above all, it is experimentally proven that only the IM-connection between layers has significant robustness, even in harsh situations such as a low-resource scenario.

Also, the substitution of IM-connection for the monotone connection implies its applicability to PLMs employing monotone connection interlayers. This integrated operation within layers effectively boosts the robustness of PLMs without increasing the number of parameters.

4 Experiment

In this section, we conduct experiments in various adversarial attacks and resource situations to demonstrate the inherent robustness of IM-BERT.

4.1 Robustness against Various Adversarial Attacks

Setup To evaluate the robustness, we utilize Adversarial GLUE (AdvGlue) (Wang et al., 2021b), a robustness benchmark. AdvGLUE comprises various adversarial attacks for some tasks of GLUE (Wang et al., 2018). In public, the dataset includes a substantial number of adversarial samples encompassing word-level, sentence-level perturbations, and human-crafted examples. In our experiments, we

N=1000	SST-2	QQP	QNLI	MNLI-m	MNLI-mm	RTE	*Avg.*
R3F	30.4	37.2	45.3	31.5	31.5	37.0	35.5
SMART	31.1	**46.2**	45.9	17.4	30.9	34.2	34.3
FreeLB	33.8	41.0	41.9	24.8	29.0	40.7	35.2
BERT	32.21	33.8	43.5	32.5	31.3	**42.4**	36.0
IM-BERT$_{T=5}$	39	41.5	47.1	**38.3**	30.9	31.3	38.0
IM-BERT$_{T=10}$	*39.4*	38.0	*52.9*	*36.9*	**36.4**	35.0	*39.8*
IM-BERT$_{T=15}$	**44.1**	*45.3*	**55**	36.1	*34.2*	36.2	**41.8**
N=500	SST-2	QQP	QNLI	MNLI-m	MNLI-mm	RTE	*Avg.*
R3F	29.7	35.9	37.8	25.6	32.1	**49.4**	35.1
SMART	35.1	34.6	41.9	24.0	33.3	40.7	35.0
FreeLB	31.8	35.9	*48.0*	20.7	30.3	*48.2*	35.8
BERT	34	32.5	43.2	25.1	**37**	47.3	36.5
IM-BERT$_{T=5}$	41	41.9	*48.0*	35.8	34.2	32.9	38.9
IM-BERT$_{T=10}$	*41.9*	**48.3**	**49.1**	**39.7**	*36.4*	39.1	**42.4**
IM-BERT$_{T=15}$	**45.9**	*44.4*	47.3	*39.1*	32.5	36.2	*40.9*

Table 2: **Results on AdvGLUE under low-resource scenarios.** We report accuracy to evaluate the performance of IM-BERT. (Upper) Results for 1,000 instances. (Bottom) Results for 500 instances. The best performance is highlighted in bold.

fine-tune IM-BERT for standard GLUE and evaluate the robustness for AdvGLUE. We employ our baseline as bert-base-uncased. Both IM-BERT and EX-BERT are warm-started on the baseline. To make a fair comparison, we configure the same hyperparameters as BERT. During fine-tuning the model with the Adam (Kingma and Ba, 2015), we set the learning rate to 2e-5 for all tasks. Additionally, we use a linear warm-up scheduler and set the batch size to 16. The standard GLUE data is split in an 8:2 ratio for each task. The model with the lowest validation loss is selected and evaluated by the GLUE metric in the hugging face. In our model, the hyperparameter iteration T and step size γ are 5 and 0.1, respectively, in the main experiment.

Result Our analysis of IM-BERT's robustness is further validated by its stable performance on AdvGLUE. When compared to EX-BERT and BERT, IM-BERT consistently outperforms, with results on validation sets showing suppression of BERT by up to 9.8%p and EX-BERT by up to 16%p in RTE. Averaged across all AdvGLUE tasks, IM-BERT exhibits a 5.4%p higher accuracy than EX-BERT. Even on test sets, IM-BERT demonstrates its stability, yielding an 8.3%p higher accuracy than BERT. Comparing IM-BERT with standard training against adversarial training, we utilize results from AdvGLUE and the CreAT paper. On test sets, IM-BERT in standard fine-tuning outperforms baseline by up to 8.33%p on average. Additionally, comparisons with other models show that IM-BERT performs similarly or superiorly to BERT$_{large}$ and adversarial training methods on BERT$_{large}$, achieving up to 9.33%p higher accuracy over SMART$_{large}$ (Jiang et al., 2020) on average. These results affirm IM-BERT's robustness against perturbations. Table 1 shows that IM-RoBERTa performs well across multiple tasks, notably achieving high scores in SST2, QQP, QNLI, and RTE. Its average performance is higher than FreeLB by up to 1.27%p and significantly higher than InfoBERT by up to 5.7%p, indicating it generally outperforms other methods. Specifically, IM-RoBERTa achieves the highest score in QQP and second-place solid finishes in other tasks, demonstrating robust overall performance. RTE and MNLI show relatively lower improvements because IM-RoBERTa focuses on robustness against adversarial perturbations rather than reasoning abilities. Consequently, IM-RoBERTa excels in tasks like QQP while maintaining competitive performance elsewhere. Compared to traditional adversarial training methods, this approach reduces the need for extensive hyperparameter tuning by relying on standard fine-tuning. Thus, the IM-connection enhances inherent robustness with simple architectural modifications.

4.2 Robustness in Low Resource Scenarios

PLMs are vulnerable to adversarial attacks due to scarce data in downstream tasks. With fewer resources, the model becomes more susceptible. IM-BERT demonstrates its robustness in this challenging scenario.

Setup To evaluate the robustness of models in harsh situations like a low-resource training scenario, we sample 1000 and 500 instances from clean GLUE as train and validation sets and evaluate the mod-

	Word	Sentence	Human	All	FLOPs / Params
BERT	30.1(±8.8)	12.2(±7.1)	24.4(±3.5)	29.1(±2.3)	22.35 / 109.48M
Layer (1-3)	**46.2(±8.0)**	29.4(±3.1)	28.4(±2.2)	37.8(±1.4)	50.29 / 109.48M
Layer (4-6)	*41.9(±1.4)*	43.1(±1.8)	**36.6(±1.9)**	**41.2(±1.6)**	
Layer (7-9)	38.1(±2.7)	*50.5(±8.2)*	26.9(±1.6)	35.8(±2.3)	
Layer (10-12)	35.5(±5.3)	28.9(±11.4)	24(±7.5)	29.3(±3.4)	
IM-BERT	39.2(±4.1)	**58.8(±13.5)**	*33.8(±6.2)*	*39.2(±0.8)*	134.1 /109.48M

Table 3: **Results on SST-2 in AdvGLUE** This table evaluates the efficacy of IM-connection against various adversarial attacks, measuring accuracy at the word, sentence, and human-crafted example levels. The "All" column provides an overall performance metric. The bold highlights the best performance. The computational efficiency of each model is indicated by FLOPs and parameters. Each result is averaged over three runs.

els with the validation set of AdvGLUE. For a fair comparison, we conduct 3 runs with the same random seed in each run and report the average result. Other settings are the same as the main experiment.

Result The limited data of downstream tasks and the complexity of PLMs result in overfitting and vulnerability against adversarial attacks. The smaller the training dataset, the greater the vulnerability to overfitting and advertising attacks. Table 2 verifies that IM-connection robustifies the network in low-resource scenarios. With a sample of 1000 instances, the accuracies of IM-BERTs are consistently higher than those of the baseline for all tasks except for RTE. In the experiment with 500 samples, our models also maintain outperformance except for MNLI-mm and the RTE task. The gap between IM-BERT and baseline achieves higher accuracy up to 11.9%p and 15.8%p, respectively. Overall, IM-BERT with $T = 10$ and $T = 15$ performs better, 5.8%p and 5.9%p over the baseline, respectively. The gaps are more significant within 500 instances than in 1000, implying that our approach effectively robustifies BERT even in low-resource situations.

4.3 Ablation Study

To illustrate the effectiveness of our method for fine-tuning, we conduct additional experiments. These results indicate the effective placement of IM-connection and provide solutions to the inherent time-cost issues of implicit methods.

How does IM-connection make BERT robust? As analyzed in the previous section, the implicit method effectively prevents the divergence of hidden states when perturbations are inserted into the initial value. To investigate how the IM-connection enhances the robustness of BERT against attacks at various levels, we construct 4 models with IM-connection applied at different positions within layers. We divide the 12 layers of BERT into 4 groups, fine-tune the models with SST-2, and test them with SST-2 in AdvGLUE. In this case, we train the models on the entire dataset and evaluate their performance, as shown in Table 1. To ensure fairness, we run 3 times and show the result as an average.

Type of Attack Level In Table 3, we analyze the effects of IM-connection and IM-BERT at various attack levels. Both methods enhance robustness by approximating the accurate hidden states of each token. They generally perform better in word-level perturbations and sentence-level adversarial attacks than in human-crafted examples, as they approximate hidden states for each token during prediction. This property leads to greater resistance to word-level adversarial attacks when the IM-connection is located in lower layers. Additionally, because PLMs learn semantics as they pass through layers, applying IM-connection to the middle layers helps mitigate perturbations, thereby preventing sentence-level attacks. Conversely, when the IM-connection is located in the middle layer, BERT becomes more robust against sentence-level adversarial attacks due to its improved ability to learn semantic representations. This is relevant to where the IM-connection conveys well-converged values on hidden states.

Location of IM-connection Table 3 reveals that placing the IM-connection in higher layers leads to poorer performance compared to lower layers. Specifically, when the IM-connection is positioned in higher layer groups like Layer (10-12), it fails to defend effectively against adversarial attacks. This suggests that BERT's earlier layers struggle to converge well on hidden states without the IM-connection, transmitting divergent values. In contrast, lower layers with the IM-connection can better approximate and deliver stable hidden states when the input is perturbed, successfully mitigat-

ing the attack. In other words, a well-approximated hidden state can be delivered via the IM-connection located in the lower layers, enhancing robustness against adversarial attacks.

Strategic Application Based on Efficiency Table 2 in the previous section shows the trade-off between time latency and robustness due to the inherent characteristics of the IM-connection. The experiment reveals that applying the IM-connection selectively to Layers (4-6) offers better performance than applying it to all layers, as seen in IM-BERT. This strategic approach, informed by an understanding of the PLM's structure, results in 2.25 times fewer FLOPs than IM-BERT, effectively addressing the time cost issue associated with the implicit method. This strategy opens up the possibility of an efficient application method that aims to maintain the number of parameters of the IM-connection while enhancing robustness. Additional analysis on BERT and RoBERTa regarding the trade-off and strategy is provided in Appendix A.

4.4 Evaluating the Inherent Robustness

Adversarial training methods require hyperparameter tuning based on the data and are challenging to adapt to various perturbations since they train on specific perturbations (Tramer and Boneh, 2019). This limitation underscores the necessity of improving a model's inherent robustness. Our additional experiments demonstrate that IM-connection enhances inherent robustness through simple standard training, making it robust against various perturbations.

Table 4 shows the average performance of RoBERTa-based models across all tasks in the AdvGLUE dataset, broken down by each attack level. The results of the RoBERTa-based experiments confirm that IM-connection provides consistent robustness against various levels of perturbation. This demonstrates that unlike adversarial training—which is tailored to specific types of perturbations—IM-connection enhances the model's inherent robustness, enabling it to effectively handle a wide range of adversarial attacks.

	Adversarial Training			Standard Training
Attack-Level	SMART	FreeLB	InfoBERT	IM-RoBERTa
Word-Level	**62.01**	52.29	55.42	*56.3*
Sentence-Level	**47.72**	44.24	38.03	*46.44*
Human Crafted	28.69	**46.12**	33.74	*41.89*
Avg.	46.14	*47.55*	42.4	**48.21**

Table 4: **Performance comparison of RoBERTa-based models on the AdvGLUE dataset across different attack levels** The table shows the average accuracy of models trained with adversarial training methods (SMART, FreeLB, InfoBERT) versus standard training with IM-connection (IM-RoBERTa). The results are broken down by attack level (Word-Level, Sentence-Level, Human Crafted) and the overall average. The best performance for each category is highlighted in bold, and the second-best is in italic

5 Conclusion

In this paper, we propose an approach to enhance PLMs with a more robust architecture by interpreting them as an ODE solver. Through our analysis of the numerical stability, we find that the implicit method exhibits absolute stability against initial value perturbations regardless of the step size γ. Leveraging this property, we introduce IM-connection using the gradient descent method to approximate the solution of the implicit method. IM-connection effectively captures well-approximated hidden states without increasing the number of parameters. Our experimental results demonstrate the effectiveness of our approach on BERT. By simply modifying BERT to IM-BERT with incorporating IM-connection, we fine-tune IM-BERT in standard training. IM-BERT becomes more resistant to various adversarial attacks, outperforming both baseline and other adversarial training methods in the AdvGLUE dataset. Furthermore, in low-resource scenarios where the model is prone to overfitting on limited data, our method showcases its effectiveness by achieving higher accuracy compared to BERT.

Limitation

Although we improve the model's intrinsic robustness, challenges remain, particularly the high time cost associated with the implicit method. Our experiments suggest ways to reduce this time cost without compromising robustness. One approach is to apply the IM-connection selectively across multiple layers, optimizing the balance between time-cost and performance for different PLM architectures. While this study focuses on BERT-based models, future work will extend to decoder-only and encoder-decoder architectures. Additionally, we will explore alternative numerical methods, such as gradient descent, to address time-cost issues more efficiently and improve robustness across various PLMs.

Ethics Statement

Our research upholds ethical standards by utilizing publicly available datasets for model fine-tuning, ensuring transparency and avoiding sensitive data. We build upon openly released pre-trained models, enhancing accessibility and transparency. By creating a model that effectively defends against adversarial attacks, we contribute to a safer and more resilient AI system. Our objective is to democratize advanced AI techniques, including our proposed approach, making them accessible to researchers across different resource levels and fostering inclusivity in the field.

Acknowledgment

This research was supported by the Basic Science Research Program (NRF-2022R1C1C1008534) and the National R&D Program (2021M3H2A1038042) through the National Research Foundation of Korea (NRF), funded by the Ministry of Education and the Ministry of Science and ICT, respectively. It was also supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) through the Korea government (MSIT) under Grant No. 2021-0-01341 (Artificial Intelligence Graduate School Program, Chung-Ang University).

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Kendall E Atkinson and Weimin Han. 1993. *Elementary numerical analysis*. Wiley New York.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Aaron Baier-Reinio and Hans De Sterck. 2020. N-ode transformer: A depth-adaptive variant of the transformer using neural ordinary differential equations. *arXiv preprint arXiv:2010.11358*.

Martin Braun and Martin Golubitsky. 1983. *Differential equations and their applications*, volume 2. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31:6571–6583.

Chengyu Dong, Liyuan Liu, Zichao Li, and Jingbo Shang. 2020. Towards adaptive residual network training: A neural-ode perspective. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2616–2626.

Eldad Haber and Lars Ruthotto. 2017. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8018–8025.

Kenji Kawaguchi. 2020. On the theory of implicit deep learning: Global convergence with implicit layers. In *International Conference on Learning Representations*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Bei Li, Quan Du, Tao Zhou, Yi Jing, Shuhan Zhou, Xin Zeng, Tong Xiao, JingBo Zhu, Xuebo Liu, and Min Zhang. 2022. Ode transformer: An ordinary differential equation-inspired model for sequence generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8335—8351.

Mingjie Li, Lingshen He, and Zhouchen Lin. 2020. Implicit euler skip connections: Enhancing adversarial robustness via numerical stability. In *Proceedings of*

the International Conference on Machine Learning, pages 5874–5883.

Jieyu Lin, Jiajie Zou, and Nai Ding. 2021. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 333–342.

Megh Thakkar Xin Li Shafiq Joty Luo Si Linlin Liu, Xingxuan Li and Lidong Bing. 2023. Towards robust low-resource fine-tuning with multi-view compressed representations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, page 4799–4816.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-yan Liu. 2020. Understanding and improving transformer from a multi-particle dynamic system point of view. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3276–3285.

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

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Viktor Reshniak and Clayton G Webster. 2020. Robust learning with implicit residual networks. *Machine Learning and Knowledge Extraction*, 3(1):34–55.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Robin M Schmidt. 2019. Recurrent neural networks (rnns): A gentle introduction and overview. *arXiv preprint arXiv:1912.05911*.

Jiawei Shen, Zhuoyan Li, Lei Yu, Gui-Song Xia, and Wen Yang. 2020. Implicit euler ode networks for single-image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 218–219.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.

Florian Tramer and Dan Boneh. 2019. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021a. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021b. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Hongqiu Wu, Yongxiang Liu, Hanwen Shi, Min Zhang, et al. 2022. Toward adversarial training on contextualized language representation. In *International Conference on Learning Representations*.

Yibo Yang, Jianlong Wu, Hongyang Li, Xia Li, Tiancheng Shen, and Zhouchen Lin. 2020. Dynamical system inspired adaptive time stepping controller for residual network families. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6648–6655.

Penghang Yin, Minh Pham, Adam Oberman, and Stanley Osher. 2018. Stochastic backward euler: an implicit gradient descent algorithm for k-means clustering. *Journal of Scientific Computing*, 77:1133–1146.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between cv and nlp! a gradient-based textual adversarial attack framework. *arXiv preprint arXiv:2110.15317*.

Jing Zhang, Peng Zhang, Baiwen Kong, Junqiu Wei, and Xin Jiang. 2021. Continuous self-attention models with neural ode networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, volume 35, pages 14393–14401.

Jingfeng Zhang, Bo Han, Laura Wynter, Bryan Kian Hsiang Low, and Mohan Kankanhalli. 2019. Towards robust resnet: A small step but a giant leap. In *Proceedings of the Twenty-Eighth International Joint Conference on Artifcial Intelligence*, pages 4285–4291.

Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. 2017. Polynet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 718–726.

Rui Zheng, Rong Bao, Qin Liu, Tao Gui, Qi Zhang, Xuan-Jing Huang, Rui Xie, and Wei Wu. 2022. Plugat: A plug and play module to defend against textual adversarial attack. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2873–2882.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

Mai Zhu, Bo Chang, and Chong Fu. 2023. Convolutional neural networks combined with runge-kutta methods. *Neural Computing and Applications*, 35(2):1629–1643.

6 Appendix

A Robustness and Time Latency with Iteration *T*

As a numerical approach, we utilize the gradient descent method to solve hidden states, causing iterative computations of *T* for each layer.
To analyze our assumption, the trade off between iteration *T* and performance, we conduct additional experiment.

A.1 Robustness with diverse Iteration *T*

We conduct experiments utilizing TextFooler (Yuan et al., 2021) and IMDB (Maas et al., 2011). We train IM-BERT on the IMDB dataset and evaluate against adversarial attack generated by TextFooler, varying *T* from 1 to 15. In Table 4, as the iteration *T* increases, the result indicates that IM-connection enhances the network's stability against adversarial attacks, with accuracy peaking at $T = 10$. When $T = 10$, the result exhibits better accuracy with fewer FLOPs than $T = 15$. This observation suggests that we can identify optimal iteration *T* that maximizes performance for the specific dataset.

TextFooler	*T*=1	*T*=5	*T*=10	*T*=15
Accuracy	0.2228	0.2316	**0.2747**	0.2633
FLOPs	44.7	134.1	245.85	357.59

Table 5: **Results on IMDB attacked by TextFooler** we report Accuracy and FLOPs to evaluate the performance according to *T* in IM-connection

A.2 Strategic IM-connection Application to mitigate time-latency issue

	Accuracy	FLOPs	Params
BERT	29.1	22.35	109.48M
SMART_{BERT}	20.6	67.09	
IM-BERT(1-3)	37.8	50.29	
IM-BERT(4-6)	**41.2**		
IM-BERT(7-9)	35.8		
IM-BERT(10-12)	29.3		
IM-BERT(All)	*39.2*	134.1	
$\text{SMART}_{RoBERTa}$	50.9	236.88	355.36M
IM-RoBERTa(4-7)	53.4	144.76	
IM-RoBERTa(11-14)	**54.7**		
IM-RoBERTa(All)	52	473.75	

Table 6: **Result on SST-2 in AdvGLUE with BERT and RoBERTa** This table evaluates the efficacy of IM-connection against adversarial attack on SST-2 in AdvGLUE. The bold highlights the best performance. The computational efficiency of each model is indicated by FLOPs and Params.

While the gradient descent approach is essential for solving the implicit method, it introduces time latency, evident from Table 4's linear increase in latency according to Algorithm 1. To mitigate this, Table 3 suggests applying IM-connection to specific layers. Incorporating IM-connection in early or middle layers alleviates time latency and ensures adversarial robustness. The following Table 6 reinforces the benefits of this strategic application.

Table 6 illustrates the effectiveness of strategically applying the IM-connection into specific layers in both BERT and RoBERTa models. The results demonstrate that selectively incorporating IM-connection in certain layers, particularly in the middle layers (e.g., Layers 4-6 for BERT and Lay-

ers 11-14 for Roberta), improves accuracy while managing computational efficiency (FLOPs) and the number of parameters.

When IM-connection is applied to all layers, IM-RoBERTa demands roughly 2.00x the FLOPs of SMART, indicating a significant increase in computational cost without a corresponding boost in adversarial robustness. However, selectively applying IM-connection to middle layers, like Layers 4-6 in BERT and 11-14 in RoBERTa, uses resources more efficiently. This selective approach requires 1.64x fewer FLOPs than the all-layer IM-RoBERTa and surpasses SMART in both robustness and accuracy. The strategic layer-specific application of IM-connection effectively balances computational efficiency with enhanced adversarial resistance. This method highlights the benefits of focusing on the most impactful layers to optimize performance.