# Is LoRA Robust to Adversarial Attacks? Expectation-Based Regularization for Parameter-Efficient Adversarial Training

**Mingze Ni[1], Yongshun Gong[2], Xinghao Yang[3], Wei Liu[1]***
[1]School of Computer Science, University of Technology Sydney, Australia
[2]School of Software, Shandong University
[3]College of Control Science and Engineering, China University of Petroleum (East China)
{mingze.ni, wei.liu}@uts.edu.au, ysgong@sdu.edu.cn, yangxh@upc.edu.cn

## Abstract

Low-Rank Adaptation (LoRA) is widely used for parameter-efficient fine-tuning, yet its robustness to textual adversarial perturbations remains understudied. Existing adversarial training methods often improve robustness by solving costly min-max objectives with inner perturbation optimization with suboptimal results. In this research, we propose **E**xpectation-**R**egularized **LoRA (ER-LoRA)**, a simple expectation-based regularization approach that improves the robustness of LoRA without requiring inner maximization or adversarial example generation. ER-LoRA smooths predictions over a LoRA-aware low-dimensional perturbation distribution and admits a closed-form curvature-aware objective via a local Gaussian approximation. It adds no trainable parameters and keeps the LoRA architecture unchanged. Experiments on adversarial NLP benchmarks and textual attacks show that ER-LoRA improves robustness while preserving clean accuracy and parameter efficiency.

## 1 Introduction

Large pretrained language models have achieved strong performance across a wide range of NLP tasks, but adapting them through full fine-tuning is increasingly expensive. Parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019; Pfeiffer et al., 2020; Li and Liang, 2021; Ben Zaken et al., 2022; Hu et al., 2022; Dettmers et al., 2023) provides a practical alternative by updating only a small number of task-specific parameters. Among PEFT methods, LoRA (Hu et al., 2022) has become particularly popular because it freezes the pretrained backbone and learns low-rank updates to selected weight matrices, substantially reducing memory and optimization costs while maintaining strong clean-task performance. However, while LoRA has been extensively studied for efficiency and clean accuracy, its adversarial robustness remains much less understood.

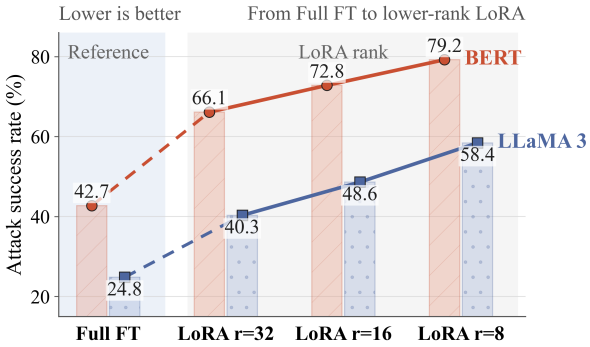



Figure 1: **LoRA shows a robustness gap under textual attacks.** We fine-tune BERT (Devlin et al., 2019) and LLaMA 3-8B (Grattafiori et al., 2024) using FFT and LoRA ranks $r \in \{8, 16, 32\}$, then evaluate PSO (Zang et al., 2020) under identical constraints. The y-axis shows average attack success rate, where lower is better. Increasing $r$ slightly improves robustness, yet LoRA remains consistently less robust than FFT on both backbones, motivating ER-LoRA.

Adversarial robustness is crucial for reliable deployment, as small semantics-preserving textual perturbations such as word substitutions can cause large prediction changes (Alzantot et al., 2018; Jin et al., 2020; Li et al., 2020; Garg and Ramakrishnan, 2020; Ebrahimi et al., 2018; Ni et al., 2024; Wang et al., 2022; Zhu et al., 2024). Although LoRA is widely used for efficient adaptation, its robustness under textual attacks remains underexplored. As an initial study, we evaluate LoRA under several word-level attacks and observe that LoRA-adapted models may experience larger robustness drops than fully fine-tuned models, while increasing the adapter rank does not necessarily bring substantial robustness gains. These findings motivate adversarially robust training for LoRA, rather than relying solely on a larger adaptation rank.

A natural direction for improving robustness is adversarial training, which trains the model against

* Corresponding author.



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 34848–34861
October 24-29, 2026 ©2026 Association for Computational Linguistics

perturbations that reveal vulnerable prediction behaviors. In computer vision, TRADES (Zhang et al., 2019) formulates robustness as a trade-off between natural accuracy and adversarial stability under a min–max objective. In NLP, adversarial training is more challenging because text is naturally discrete and valid semantics-preserving perturbations, such as word substitutions, are difficult to characterize and optimize. Existing NLP methods therefore often move the perturbation process to continuous spaces: FreeLB (Zhu et al., 2020) performs iterative adversarial updates in the embedding space, while SMART (Jiang et al., 2020) improves fine-tuning robustness by enforcing local smoothness around input examples. Although effective, these methods typically require expensive inner optimization or depend on generated textual adversarial examples with limited coverage. This is especially problematic for LoRA, whose main appeal is efficient adaptation with few trainable parameters. Directly applying conventional adversarial training may offset LoRA's efficiency advantage and may not fully exploit its low-rank adaptation structure. This motivates the need to study adversarially robust training methods designed specifically for LoRA-adapted LLMs and NLP models.

These challenges suggest an alternative to worst-case optimization: robust LoRA training may benefit from averaging over informative local perturbations rather than focusing on a single adversarial direction. Based on this intuition, we propose **E**xpectation-**R**egularized **LoRA (ER-LoRA)**, a robust parameter-efficient fine-tuning method that regularizes LoRA by minimizing the expected KL discrepancy between clean and locally perturbed predictions. Unlike adversarial methods that rely on the **maximum** discrepancy induced by worst-case perturbations, ER-LoRA optimizes this **expectation** over a LoRA-aware low-dimensional perturbation distribution, encouraging stable predictions on average around each training example. The distribution emphasizes prediction-sensitive local directions while staying within a small neighborhood, thereby injecting useful adversarial information without expensive high-dimensional inner maximization. By further modeling this distribution with a Gaussian approximation, we obtain a closed-form expectation objective, avoiding costly Monte-Carlo simulation over many perturbation samples and reducing computational overhead. ER-LoRA thus improves LoRA robustness without adding trainable parameters or modifying the original LoRA architecture.

Our contributions are summarized as follows:

- We propose **E**xpectation-**R**egularized **LoRA (ER-LoRA)**, which improves robust LoRA fine-tuning by minimizing an expected KL regularization loss over LoRA-aware low-dimensional perturbations.
- We derive a closed-form approximation to the expected robustness objective under a local Gaussian perturbation model. This yields an efficient and geometry-aware objective that avoids costly Monte-Carlo sampling and iterative inner maximization.
- We evaluate ER-LoRA on multiple NLP benchmarks and textual adversarial attacks. Results show that ER-LoRA improves robustness while preserving clean accuracy, parameter efficiency, and low training overhead.

**Related Work** Our work builds on parameter-efficient fine-tuning (PEFT), which adapts pre-trained language models by updating only a small set of task-specific parameters (Pfeiffer et al., 2020; Li and Liang, 2021; Ben Zaken et al., 2022; Rücklé et al., 2021). LoRA has become a standard PEFT method by learning low-rank weight updates, and recent variants further improve memory efficiency and optimization behavior (Hu et al., 2022; Dettmers et al., 2023; Liu et al., 2024). However, most LoRA-based studies focus on clean-data adaptation, leaving its robustness under textual perturbations less understood.

Adversarial robustness in NLP differs from vision because text is discrete and valid perturbations must preserve fluency and semantics (Morris et al., 2020; Li and Qiu, 2021). Existing studies therefore follow two main directions. Textual attacks construct adversarial examples through character edits, word substitutions, or paraphrases to test model vulnerability under meaning-preserving input changes (Morris et al., 2020; Wang et al., 2022; Zhu et al., 2024; Ni et al., 2024). Robust training methods instead often perturb continuous embeddings or hidden states to avoid direct discrete optimization (Liu et al., 2020; Chivukula et al., 2021; Li and Qiu, 2021). Classical adversarial training searches for worst-case perturbations by optimizing a min-max objective (Goodfellow et al., 2015; Yin et al., 2018; Madry et al., 2018), while discrepancy-based methods such as TRADES (Zhang et al., 2019) and

virtual adversarial training promote local prediction smoothness through consistency regularization (Miyato et al., 2018). In pretrained language model fine-tuning, FreeLB (Zhu et al., 2020) performs iterative adversarial updates in embedding space, SMART (Jiang et al., 2020) enforces smoothness around each input, and R3F (Aghajanyan et al., 2021b) injects random representation perturbations. Other NLP objectives further combine embedding-space perturbations with token-aware normalization or contrastive representation alignment (Li and Qiu, 2021; Pan et al., 2022). These methods provide useful robustness signals, but they are not tailored to LoRA's low-rank update structure and can weaken the efficiency advantage of PEFT. ER-LoRA instead regularizes expected prediction discrepancies over a LoRA-aware low-dimensional perturbation distribution.

## 2 Background

For an input-label pair $(x, y) \sim \mathcal{D}$, adversarial attacks aim to find a small perturbation $\delta$ that maximizes the model loss while remaining inside a prescribed neighborhood of the original input. This perturbation is usually constrained by a norm bound, such as $\|\delta\|_p \le \epsilon$, and can be written as

$$\delta^\star = \arg\max_{\|\delta\|_p \le \epsilon} \ell_{\text{ce}}(p_\theta(x+\delta), y)\,, \qquad (1)$$

where $p_\theta(\cdot)$ denotes the predictive distribution and $\ell_{\text{ce}}$ is the cross-entropy loss.

Adversarial training improves robustness by optimizing the model on such worst-case perturbed examples. The standard objective is formulated as a min-max problem:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[ \max_{\|\delta\|_p \le \epsilon} \ell_{\text{ce}}(p_\theta(x+\delta), y) \right]. \qquad (2)$$

The inner maximization searches for adversarial examples, while the outer minimization updates the model parameters to reduce the adversarial risk.

Another common formulation is discrepancy-based robust training. For example, TRADES (Zhang et al., 2019) replaces the adversarial cross-entropy term with a KL divergence between the clean and perturbed predictions:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \Big[ \ell_{\text{ce}}(p_\theta(x), y) + \beta \max_{\|\delta\|_p \le \epsilon} D_{\text{KL}}\left(p_\theta(x) \,\|\, p_\theta(x+\delta)\right) \Big], \qquad (3)$$

where $\beta > 0$ balances natural accuracy and robustness. This KL term encourages local prediction smoothness by making the model output stable under small perturbations.

## 3 Expectation-Based Regularizer for LoRA

ER-LoRA smooths LoRA fine-tuning by penalizing the *expected* KL deviation under task-relevant hidden perturbations, rather than optimizing against a single worst-case perturbation.

### 3.1 Local Perturbation Model

Consider a LoRA-adapted linear layer with effective transformation $W_{\text{eff}} h = Wh + BAh$, where $A \in \mathbb{R}^{r\times d}$, $B \in \mathbb{R}^{d_{\text{out}}\times r}$, and $h \in \mathbb{R}^d$ is the input representation to this layer. Let $f_\theta(h) \in \mathbb{R}^C$ denote the logits produced by the remainder of the network and $p_\theta(h) = \text{softmax}(f_\theta(h))$ the predictions. We perturb $h$ within an $r$-dimensional subspace:

$$h' = h + Uz, \qquad U \in \mathbb{R}^{d\times r},\ U^\top U = I_r,\ r \ll d.$$

Rather than optimizing a single worst-case $z$, ER-LoRA enforces smoothness over a local perturbation distribution $q(z \mid h)$:

$$R_{\text{ER}}(\theta; h) = \mathbb{E}_{z\sim q(\cdot|h)}\big[ D_{\text{KL}}(p_\theta(h) \,\|\, p_\theta(h+Uz))\big], \qquad (4)$$

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)}\big[\ell_{\text{ce}}(p_\theta(h), y) + \lambda\, R_{\text{ER}}(\theta; h)\big], \qquad (5)$$

where $\lambda > 0$ is the regularization weight.

### 3.2 LoRA-aware Perturbation Subspace

A LoRA-adapted layer acts as $W_{\text{eff}} h = Wh + BAh$, so the adapter branch depends on $h$ only through $Ah$. Any component of $h$ orthogonal to $\text{row}(A)$ is invisible to the trainable factors and cannot be compensated by gradient updates on $A, B$; regularizing there wastes capacity. We therefore take

$$U_A = \text{orth}(A^\top) \qquad (6)$$

as the intrinsic, parameter-induced part of the subspace.

Although $U_A$ reflects the current LoRA subspace, it does not use the supervised signal from the mini-batch. To include task-sensitive directions, we augment $U_A$ with a batch-level loss-increasing

direction. Let $g_i = \nabla_{h_i} \ell_{\mathrm{ce}}(p_\theta(h_i), y_i)$. For a mini-batch of size $N$, we define

$$\bar{g} = \frac{1}{N} \sum_{i=1}^{N} \frac{g_i}{\|g_i\|_2 + \epsilon}, \qquad U = \mathrm{orth}([\bar{g}, U_A]). \tag{7}$$

Thus the perturbation dimension satisfies $r \leq r_L + 1$. The normalization prevents a few large-gradient examples from dominating the shared direction, and only affects the construction of $U$.

### 3.3 Gaussian Perturbation and Closed-form Regularizer

Let $F(h) = \ell_{\mathrm{ce}}(p_\theta(h), y)$ denote the supervised loss at the hidden representation $h$. We define an energy over local coordinates $z$ as

$$\Phi(z; h) = F(h + Uz) - \frac{\tau}{2}\|z\|_2^2, \tag{8}$$

where $\tau > 0$ controls the locality of the perturbation. Since $U^\top U = I_r$, the quadratic penalty in $z$ is equivalent to an $\ell_2$ penalty in the hidden space. The induced perturbation distribution is

$$q(z \mid h) \propto \exp(\Phi(z; h)). \tag{9}$$

We use this formal local distribution to define the target perturbation behavior: it favors directions that increase the task loss, while the quadratic term keeps perturbations near $h$.

**Gaussian approximation.** Sampling from Eq. (9) is generally intractable. We therefore derive a local Gaussian approximation by expanding the function $z \mapsto F(h + Uz)$ around $z = 0$, with $U$ fixed. Let

$$g = \nabla_h F(h), \qquad H = \nabla_h^2 F(h) \tag{10}$$

denote the gradient and Hessian of the supervised loss at $h$. Then

$$F(h + Uz) \approx F(h) + g^\top Uz + \frac{1}{2} z^\top U^\top H U z. \tag{11}$$

Substituting Eq. (11) into Eq. (8) gives

$$\Phi(z; h) \approx F(h) + b^\top z - \frac{1}{2} z^\top \Lambda z, \tag{12}$$

where

$$b = U^\top g, \tag{13}$$

$$\Lambda = \tau I_r - U^\top H U. \tag{14}$$

Here, $b$ is the projected loss gradient and $\Lambda$ is the precision matrix induced by the locality penalty and the projected loss curvature.

**Theorem 1** (Local Gaussian perturbation model)**.** *If $\Lambda \succ 0$, then under the local quadratic approximation, $q(z \mid h) \approx \mathcal{N}(\mu, \Sigma)$, where*

$$\mu = \Lambda^{-1} b, \qquad \Sigma = \Lambda^{-1}. \tag{15}$$

We next approximate the expected prediction discrepancy under this local Gaussian perturbation. Define the KL curvature in the perturbation coordinates as

$$A_\theta = \nabla_z^2 D_{\mathrm{KL}}(p_\theta(h) \,\|\, p_\theta(h + Uz))\Big|_{z=0}. \tag{16}$$

**Corollary 1** (Local robustness decomposition)**.** *Let the local Gaussian approximation be $q(z \mid h) \approx \mathcal{N}(\mu, \Sigma)$, with $\mu$ and $\Sigma$ given by Theorem 1. Under a second-order approximation of the KL divergence around $z = 0$,*

$$R_{\mathrm{ER}}(\theta; h) \approx \frac{1}{2} \mathrm{tr}(A_\theta \Sigma) + \frac{1}{2} \mu^\top A_\theta \mu. \tag{17}$$

Proofs are provided in Appendix A. The trace term penalizes the average local curvature of the prediction distribution within the stochastic neighborhood specified by $\Sigma$, while the quadratic term penalizes sensitivity along the mean loss-increasing direction $\mu$.

**Practical curvature approximation.** The closed-form regularizer requires the projected loss Hessian $U^\top H U$ and the KL curvature $A_\theta$. To avoid materializing the full Hessian, we compute curvature directly in the subspace spanned by $U$. Let

$$J = \nabla_h f_\theta(h), \qquad Z = JU. \tag{18}$$

We approximate the KL curvature by the projected Fisher/Gauss–Newton matrix

$$\widehat{A}_\theta = Z^\top \mathcal{F}_{p_\theta(h)} Z. \tag{19}$$

We also use this matrix as a positive-semidefinite surrogate approximation for the projected loss Hessian $U^\top H U$. To obtain a valid Gaussian precision matrix, we define the spectrally shifted precision

$$\widehat{\Lambda} = \Big(\lambda_{\max}(\widehat{A}_\theta) + \tau_0\Big) I_r - \widehat{A}_\theta, \qquad \tau_0 > 0, \tag{20}$$

This is equivalent to choosing the adaptive locality parameter $\tau = \lambda_{\max}(\widehat{A}_\theta) + \tau_0$ in the practical precision $\tau I_r - \widehat{A}_\theta$ (Marquardt, 1963; Martens, 2020, 2010). Because $\widehat{A}_\theta \succeq 0$, the eigenvalues of $\widehat{\Lambda}$ are $\lambda_{\max}(\widehat{A}_\theta) - \lambda_j(\widehat{A}_\theta) + \tau_0$, and hence are all

**Algorithm 1** ER-LoRA training step

**Require:** Mini-batch $\{(x_i, y_i)\}_{i=1}^{N}$; LoRA factor $A$; regularization weight $\lambda$; locality parameter $\tau_0 > 0$.

Forward pass to compute $\ell_{\text{ce}}(p_\theta(h_i), y_i)$, $h_i$, and $p_\theta(h_i)$ for each example.
Compute $g_i = \nabla_{h_i} \ell_{\text{ce}}(p_\theta(h_i), y_i)$.
Build the perturbation basis $U$ using Eqs. (6)–(7).
**for** $i = 1, \ldots, N$ **do**
    Compute $b_i$, $\widehat{A}_{\theta,i}$, $\widehat{\Lambda}_i$, $\widehat{\Sigma}_i$, and $\widehat{\mu}_i$ using Eqs. (13) and (19)–(21).
    Compute $R_{\text{ER}}(\theta; h_i)$ using Eq. (22).
**end for**
Update the trainable parameters using Eq. (5).

lower-bounded by $\tau_0 > 0$. Therefore, $\widehat{\Lambda} \succ 0$, making the local Gaussian approximation well-defined. This spectral shift plays the same stabilizing role as damping in Gauss–Newton or trust-region methods. It yields a stable practical approximation rather than an exact Hessian evaluation.

We then compute the covariance and mean as

$$\widehat{\Sigma} = \widehat{\Lambda}^{-1}, \qquad \widehat{\mu} = \widehat{\Sigma} b. \tag{21}$$

The practical ER-LoRA regularizer is then

$$R_{\text{ER}}(\theta; h) \approx \frac{1}{2} \operatorname{tr}\left(\widehat{A}_\theta \widehat{\Sigma}\right) + \frac{1}{2} \widehat{\mu}^\top \widehat{A}_\theta \widehat{\mu}. \tag{22}$$

The full training procedure is summarized in Algorithm 1.

## 4 Experiments

We evaluate ER-LoRA on AdvGLUE across encoder and decoder-only backbones, focusing on robustness, clean accuracy, and efficiency.

**Datasets and Metrics.** We evaluate ER-LoRA on AdvGLUE (Wang et al., 2021), which contains adversarial versions of SST-2 (Socher et al., 2013), QQP (Wang et al., 2018), MNLI (Williams et al., 2018), and QNLI (Demszky et al., 2018). We report clean accuracy and adversarial accuracy. The AdvGLUE dataset contains pre-generated adversarial examples at the sentence level (e.g., T3(Wang et al., 2020), AdvFever(Thorne et al., 2019), and SCPN(Iyyer et al., 2018)), at the word level (e.g., TextFooler(Jin et al., 2020), BERT-Attack(Li et al., 2020), and PSO(Zang et al., 2020)), as well as human-generated adversarial examples. Additional dataset and evaluation details are provided in Appendix B.

**Implementation details.** Our main experiments use RoBERTa-base (Liu et al., 2019) and DeBERTa-base (He et al., 2021) as backbone models. We compare ER-LoRA with LoRA and robust LoRA fine-tuning baselines, including FreeLB-LoRA (Zhu et al., 2020), SMART-LoRA (Jiang et al., 2020), TRADES-LoRA, UIAT-LoRA (Dong et al., 2023), and PWWS-AT-LoRA (Ren et al., 2019). Unless otherwise specified, all methods use LoRA rank $r = 8$ and apply LoRA modules to attention projection matrices. Additional implementation details, hyperparameter settings, and results for all backbones are provided in Appendix B.

### 4.1 Main Results

We analyze the main results along four axes: adversarial robustness, training accuracy and stability, computational efficiency, and the trade-off between clean accuracy and adversarial robustness.

**Adversarial robustness** Table 1 shows that ER-LoRA delivers stronger adversarial robustness than standard LoRA and most robust-training baselines, especially on RoBERTa-base. Figure 2 further shows that this advantage appears early in training and is maintained across epochs. These results indicate that ER-LoRA improves the robustness of LoRA fine-tuning without relying on a larger adapter rank or costly adversarial example generation. We attribute the gains to the expectation-based regularizer, which smooths predictions over task-sensitive perturbations in the LoRA update space and therefore exposes the model to more realistic local variations during training.

**Training accuracy and stability** Table 1 shows that ER-LoRA keeps clean accuracy close to standard LoRA while improving adversarial accuracy. This suggests that the robustness improvement does not come at the cost of clean-task performance. Figure 2 further shows that ER-LoRA converges quickly in cross-entropy loss and maintains stable adversarial accuracy after the early training stage. We attribute this behavior to the regularizer's perturbation distribution, which approximates realistic noise around the learned representation, strengthens task-relevant features, and provides a smoother optimization signal. As a result, ER-LoRA can reach strong clean performance, accelerate convergence, and avoid large oscillations in adversarial accuracy during training.

Table 1: Clean and adversarial (*Adv*) accuracy (%) on GLUE and AdvGLUE (Wang et al., 2021) validation sets. Clean/Adv stacked per dataset, mean$_{\pm}$SD over 5 different seeds. All methods use LoRA fine-tuning from the same pretrained backbone; only the regularization objective differs. For each adversarial accuracy column, **bold** indicates the best result and *italic* indicates the second-best result.

| | *RoBERTa-base* | | | | *DeBERTa-base* | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | SST-2 *Clean* / *Adv* | QNLI *Clean* / *Adv* | QQP *Clean* / *Adv* | MNLI *Clean* / *Adv* | SST-2 *Clean* / *Adv* | QNLI *Clean* / *Adv* | QQP *Clean* / *Adv* | MNLI *Clean* / *Adv* |
| LoRA ($r$=8) | 92.6$_{\pm 2.0}$ / 39.2$_{\pm 0.9}$ | *90.2*$_{\pm 2.2}$ / 48.6$_{\pm 3.0}$ | 87.6$_{\pm 2.3}$ / 32.6$_{\pm 1.4}$ | 84.8$_{\pm 3.7}$ / 31.3$_{\pm 2.5}$ | 94.4$_{\pm 1.7}$ / 59.7$_{\pm 1.1}$ | **95.4**$_{\pm 2.7}$ / *65.2*$_{\pm 4.3}$ | 89.4$_{\pm 2.4}$ / 59.2$_{\pm 2.7}$ | *90.7*$_{\pm 3.2}$ / *63.3*$_{\pm 1.7}$ |
| LoRA ($r$=16) | **96.1**$_{\pm 2.6}$ / 34.5$_{\pm 2.5}$ | 89.0$_{\pm 1.5}$ / 47.0$_{\pm 1.3}$ | 88.0$_{\pm 1.9}$ / 34.0$_{\pm 2.4}$ | 85.5$_{\pm 3.1}$ / 34.5$_{\pm 3.3}$ | **96.7**$_{\pm 3.3}$ / 59.7$_{\pm 1.9}$ | 94.1$_{\pm 4.4}$ / 64.1$_{\pm 4.0}$ | 87.3$_{\pm 4.2}$ / 61.8$_{\pm 4.8}$ | **91.5**$_{\pm 2.1}$ / 61.8$_{\pm 1.7}$ |
| LoRA ($r$=32) | 94.5$_{\pm 1.0}$ / 34.8$_{\pm 3.3}$ | 88.4$_{\pm 1.7}$ / 46.1$_{\pm 1.9}$ | 88.2$_{\pm 1.1}$ / 31.8$_{\pm 1.5}$ | **86.7**$_{\pm 4.0}$ / 35.2$_{\pm 3.3}$ | *96.2*$_{\pm 1.6}$ / *62.8*$_{\pm 2.0}$ | 94.0$_{\pm 2.9}$ / 59.5$_{\pm 2.2}$ | 89.4$_{\pm 1.7}$ / 62.7$_{\pm 1.7}$ | 90.6$_{\pm 1.3}$ / 60.8$_{\pm 2.1}$ |
| SMART | 90.7$_{\pm 1.3}$ / 41.1$_{\pm 2.7}$ | **92.3**$_{\pm 0.9}$ / 46.6$_{\pm 2.8}$ | 88.5$_{\pm 1.2}$ / 35.2$_{\pm 2.4}$ | 83.6$_{\pm 2.9}$ / 33.6$_{\pm 2.4}$ | 95.0$_{\pm 3.7}$ / 58.0$_{\pm 2.4}$ | *95.4*$_{\pm 3.1}$ / 62.8$_{\pm 3.1}$ | 88.9$_{\pm 2.2}$ / *64.3*$_{\pm 2.7}$ | 90.2$_{\pm 2.9}$ / 59.6$_{\pm 1.7}$ |
| TRADES | 92.4$_{\pm 1.1}$ / *41.2*$_{\pm 1.1}$ | 88.8$_{\pm 3.3}$ / *55.4*$_{\pm 2.0}$ | 88.4$_{\pm 3.4}$ / 35.7$_{\pm 2.2}$ | 82.9$_{\pm 1.4}$ / 29.6$_{\pm 2.2}$ | 96.1$_{\pm 2.5}$ / 58.3$_{\pm 1.4}$ | 93.2$_{\pm 2.2}$ / 63.8$_{\pm 1.3}$ | 89.3$_{\pm 1.1}$ / 61.7$_{\pm 1.5}$ | 88.5$_{\pm 3.7}$ / 60.9$_{\pm 3.8}$ |
| UIAT | 91.4$_{\pm 2.3}$ / 40.1$_{\pm 1.3}$ | 88.4$_{\pm 1.6}$ / 50.9$_{\pm 2.0}$ | 88.2$_{\pm 2.5}$ / *37.2*$_{\pm 2.0}$ | 83.4$_{\pm 1.9}$ / 31.8$_{\pm 1.2}$ | 95.4$_{\pm 2.0}$ / 61.8$_{\pm 1.8}$ | 91.8$_{\pm 4.0}$ / 64.0$_{\pm 1.3}$ | 89.0$_{\pm 2.7}$ / 60.5$_{\pm 2.4}$ | 89.9$_{\pm 3.7}$ / 62.0$_{\pm 0.8}$ |
| FreeLB | 90.0$_{\pm 3.0}$ / 40.5$_{\pm 2.0}$ | 89.0$_{\pm 2.2}$ / 51.0$_{\pm 1.4}$ | *89.1*$_{\pm 1.8}$ / 33.4$_{\pm 2.4}$ | 84.6$_{\pm 3.5}$ / *36.5*$_{\pm 2.0}$ | 93.2$_{\pm 2.6}$ / 62.1$_{\pm 1.1}$ | 92.8$_{\pm 2.0}$ / 64.6$_{\pm 1.9}$ | *89.6*$_{\pm 2.3}$ / 62.8$_{\pm 1.4}$ | 88.4$_{\pm 1.2}$ / 60.5$_{\pm 2.4}$ |
| PWWS-AT | *95.4*$_{\pm 4.1}$ / 38.6$_{\pm 1.0}$ | 86.7$_{\pm 1.5}$ / 50.1$_{\pm 2.2}$ | 88.1$_{\pm 3.3}$ / 35.4$_{\pm 2.4}$ | 84.3$_{\pm 2.4}$ / 33.7$_{\pm 1.3}$ | 95.1$_{\pm 1.2}$ / 60.8$_{\pm 3.1}$ | 94.8$_{\pm 3.6}$ / 64.2$_{\pm 4.2}$ | 89.1$_{\pm 3.6}$ / 63.5$_{\pm 2.3}$ | 90.2$_{\pm 2.6}$ / 61.8$_{\pm 1.5}$ |
| **ER-LoRA** | 91.9$_{\pm 2.8}$ / **45.2**$_{\pm 3.7}$ | 88.9$_{\pm 2.4}$ / **57.0**$_{\pm 1.5}$ | **89.3**$_{\pm 2.0}$ / **39.1**$_{\pm 2.3}$ | *85.8*$_{\pm 1.1}$ / **38.9**$_{\pm 1.3}$ | 93.6$_{\pm 2.4}$ / **64.5**$_{\pm 1.9}$ | 93.3$_{\pm 1.9}$ / **69.3**$_{\pm 1.6}$ | **90.9**$_{\pm 1.7}$ / **65.7**$_{\pm 1.5}$ | 90.5$_{\pm 3.4}$ / **65.5**$_{\pm 1.0}$ |

**Training Efficiency** We further examine whether the robustness gains of ER-LoRA come with additional computational cost. As shown in Figure 3, ER-LoRA introduces almost no extra training-time or memory overhead compared with standard LoRA. In contrast, conventional adversarial training baselines require substantially higher computation due to offline adversarial example generation or iterative perturbation optimization. These results suggest that ER-LoRA preserves the efficiency advantage of parameter-efficient fine-tuning while improving robustness. We attribute this efficiency to the closed-form regularizer, which avoids costly inner-loop optimization and sampling.

**Trade-off between accuracy and adversarial robustness** We further examine whether the robustness gains of ER-LoRA require a large sacrifice in clean accuracy, a common concern in robust learning (Tsipras et al., 2019; Nakkiran, 2019). Table 1 shows that ER-LoRA improves adversarial accuracy while keeping clean accuracy close to standard LoRA and competitive with robust-training baselines. This suggests that ER-LoRA achieves a favorable accuracy-robustness trade-off rather than simply shifting performance from clean examples to adversarial ones. We attribute this behavior to the expectation-based regularizer, which encourages local prediction stability around task-relevant perturbations while preserving the supervised cross-entropy objective for clean-data fitting.

## 4.2 Evaluation on LLMs

We further evaluate ER-LoRA on compact decoder-only LLMs, including Gemma-4-E2B (Google DeepMind, 2026) and Qwen3.5-0.8B (Qwen Team, 2026). As shown in Table 2, ER-LoRA obtains the best adversarial accuracy on all evaluated tasks for both decoder-only models, while keeping clean accuracy comparable to SMART and TRADES. The gains are consistent across the larger Gemma model and the smaller Qwen model, suggesting that the proposed regularizer is not tied to encoder-only architectures. By operating in the LoRA update space, ER-LoRA transfers the same robustness signal to decoder-only backbones.

## 4.3 Ablation Study

### 4.3.1 Regularization Weight

We study the effect of the regularization weight in Figure 4. The adversarial accuracy generally improves as the weight increases from zero to a

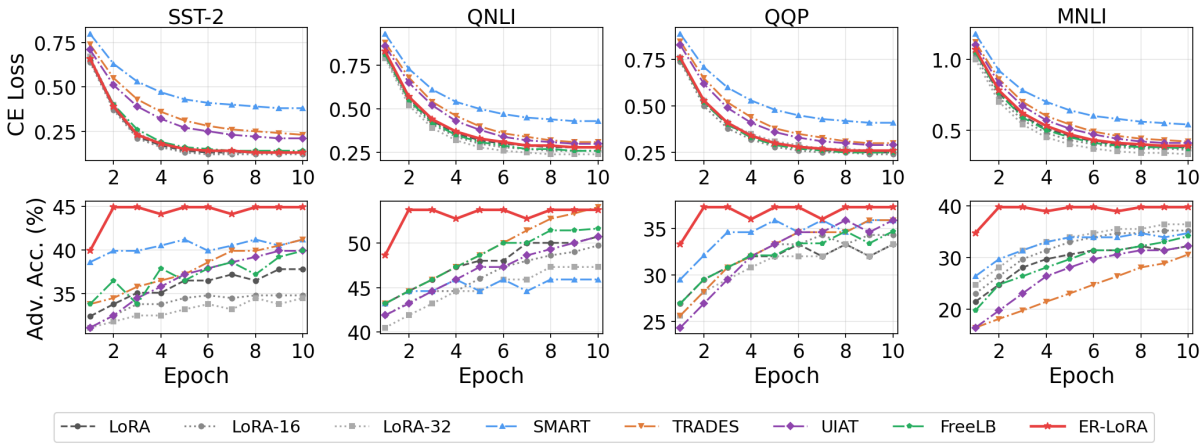



Figure 2: Training curves (CE loss and adversarial accuracy vs. epoch) on RoBERTa-base across four AdvGLUE tasks. ER-LoRA achieves consistently higher adversarial accuracy than most baselines throughout training while maintaining comparable CE loss convergence. DeBERTa-base curves are provided in Appendix C.

Table 2: Decoder-only LLM results on AdvGLUE. We compare ER-LoRA with TRADES and SMART on compact decoder-only models. Clean and Adv denote accuracy on GLUE and AdvGLUE validation splits, respectively.

| Model | Method | Clean accuracy | | | | Adversarial accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | SST-2 | QNLI | QQP | MNLI | SST-2 | QNLI | QQP | MNLI |
| Gemma-4-E2B | Zero-shot | 93.2 | 87.1 | 85.3 | 82.4 | 62.1 | 56.8 | 58.4 | 54.7 |
| | TRADES | 93.8 | 88.4 | 86.7 | 83.9 | 67.3 | 62.4 | 64.1 | 60.8 |
| | SMART | 94.0 | 88.9 | 87.2 | 84.3 | 68.5 | 63.7 | 65.3 | 62.1 |
| | **ER-LoRA** | 93.9 | **89.2** | **87.5** | **84.6** | **69.8** | **64.9** | **66.7** | **63.4** |
| Qwen3.5-0.8B | Zero-shot† | 91.3 | 84.2 | 81.6 | 78.9 | 57.2 | 52.1 | 53.8 | 50.4 |
| | TRADES | 91.8 | 85.6 | 83.2 | 80.4 | 62.8 | 58.3 | 59.7 | 56.2 |
| | SMART | 92.1 | 86.1 | 83.8 | 81.2 | 64.1 | 59.8 | 61.2 | 57.8 |
| | **ER-LoRA** | 92.0 | **86.4** | **84.1** | **81.5** | **65.7** | **61.3** | **62.8** | **59.4** |

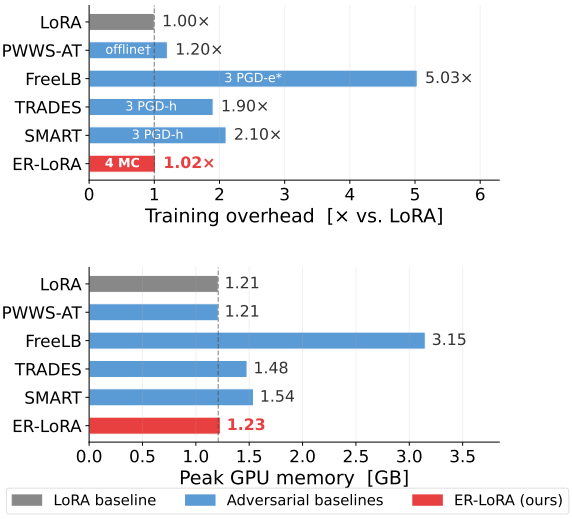



Figure 3: Training-time and memory efficiency on RoBERTa-base / SST-2. Time is normalized to standard LoRA; memory is peak GPU allocation.

moderate value, while the clean accuracy remains close to the standard LoRA setting. When the regularization becomes too strong, however, both clean and adversarial performance can decline. These results show that the proposed regularizer is beneficial when it provides sufficient local smoothing, but excessive smoothing may over-constrain the task-specific representation. We therefore use $\lambda_{\text{adv}} = 1.0$ as the default setting in the main experiments.

### 4.3.2 Closed-form Expectation

Although ER-LoRA uses a closed-form approximation for the expected regularizer, we also test whether this computation is consistent with a direct Monte Carlo estimate. Table 3 compares the closed-form objective with simulated ER, where $K$ perturbations are sampled from the Gaussian distribution in Theorem 1. The experiments were carried out independently, which present different results with Table 1. From the results in Table 3, the simulated estimates approach the closed-form result as $K$ increases, and the adversarial accuracy becomes

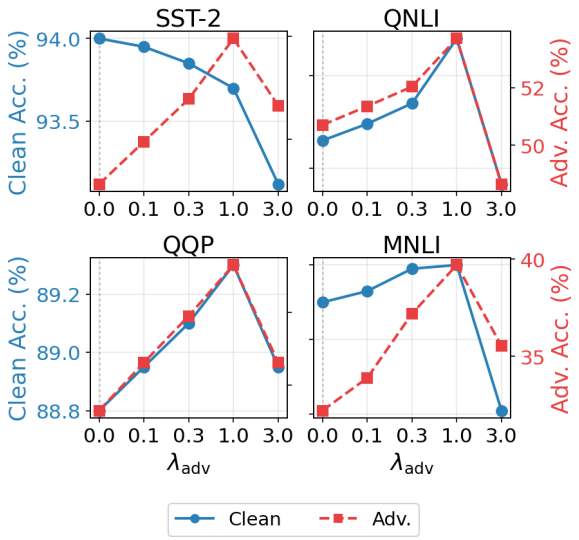



Figure 4: Effect of regularization weight $\lambda_{\text{adv}}$ on clean and adversarial accuracy for RoBERTa-base across four AdvGLUE tasks.

Table 3: Closed-form ER-LoRA vs. Monte Carlo simulation. We compare closed-form ER-LoRA with simulated expectation estimates on DeBERTa-base / SST-2, where simulated ER samples $K$ perturbations from the Gaussian distribution in Theorem 1. Overhead is normalized by closed-form ER-LoRA.

| Method | $K$ | Clean | Adv | Overhead |
|---|---|---|---|---|
| Closed-form ER | – | 94.7 | 56.8 | 1.0× |
| Simulated ER | 10 | 94.6 | 56.1 | 1.45× |
| Simulated ER | 20 | 94.7 | 56.6 | 1.85× |
| Simulated ER | 50 | 94.7 | 56.8 | 3.10× |
| Simulated ER | 100 | 94.8 | 56.9 | 5.25× |

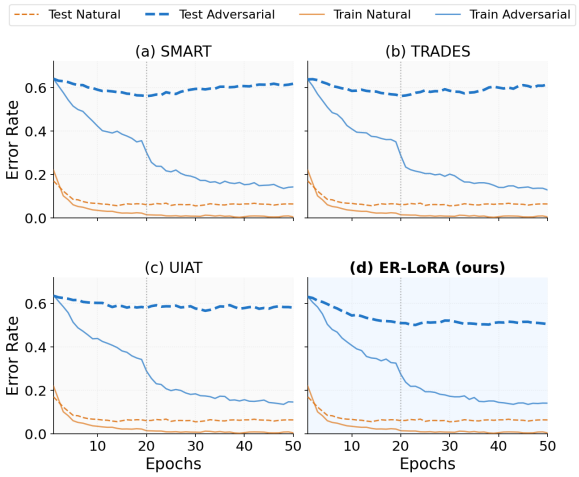



Figure 5: Robust overfitting on SST-2 with RoBERTa-base. Dashed and solid curves denote test and training error rates; the vertical dotted line marks the learning-rate drop.

similar once $K$ is moderately large. The key difference is computational cost. Monte Carlo simulation requires multiple perturbed forward passes for each example, so the overhead grows quickly with $K$. In contrast, the closed-form objective reaches nearly the same clean and adversarial accuracy with substantially lower overhead. This supports our use of the closed-form ER regularizer: it provides a reliable estimate of the expectation while preserving the efficiency advantage of LoRA fine-tuning.

Table 4: Rank ablation on RoBERTa-base / MNLI. We report clean and adversarial accuracy on AdvGLUE-MNLI under different LoRA ranks $r$.

| Rank $r$ | SMART | | TRADES | | **ER-LoRA (ours)** | |
|---|---|---|---|---|---|---|
| | Clean | Adv | Clean | Adv | Clean | Adv |
| 4 | 82.1 | 31.8 | 81.5 | 27.5 | 82.0 | **36.9** |
| 8 | 83.9 | 34.7 | 83.4 | 30.6 | 83.8 | **39.7** |
| 12 | 84.2 | 35.5 | 83.7 | 31.5 | 84.1 | **40.4** |
| 16 | 84.4 | 36.0 | 84.0 | 32.1 | 84.2 | **40.8** |
| 32 | 84.7 | 36.5 | 84.3 | 32.7 | 83.6 | **41.1** |
| 64 | 85.0 | 36.8 | 84.6 | 33.1 | 85.7 | **41.2** |

#### 4.3.3 LoRA Rank

Table 4 evaluates LoRA ranks from $r = 4$ to $r = 64$ under the same RoBERTa-base/MNLI setting. All methods benefit from increasing rank, consistent with prior observations that low-dimensional adaptation can be effective for fine-tuning (Hu et al., 2022; Li et al., 2018; Aghajanyan et al., 2021a), but the gains gradually saturate at larger ranks. ER-LoRA consistently achieves stronger clean and adversarial accuracy than SMART and TRADES across the full rank range, including the low-rank setting. These results indicate that ER-LoRA is not simply relying on extra adapter capacity. Instead, the expectation-based regularizer provides a more effective robustness signal within the same parameter budget by aligning local perturbations with the LoRA update space.

#### 4.3.4 Effectiveness of the Subspace Projection

Our method injects Gaussian noise into the hidden states at the LoRA layers to improve adversarial robustness. A key component is the subspace projection matrix $U$ in Eq. 7, which constrains the injected perturbations to directions relevant to the LoRA update space. This design enables the learnable matrix $A$ to capture additional robustness-related information while preventing perturbations from falling into orthogonal, potentially ineffective directions. To evaluate the contribution of this projection, we compare our full method with

Table 5: Ablation study of the subspace projection **U** using RoBERTa-base on SST-2. Clean and Adv. denote clean and adversarial accuracy (%), respectively. The best results are shown in bold.

| Method | Clean Acc. | Adv. Acc. |
|---|---|---|
| LoRA | **93.1** | 40.1 |
| local smoothing | 92.9 | 37.1 |
| Gradient-only | 89.5 | 41.8 |
| Random projection | 92.4 | 38.7 |
| **Ours (full projection)** | 91.5 | **47.2** |

several variants, including random-subspace projection, gradient-only perturbation($U = \mathrm{orth}(\bar{g})$), generic local smoothing($U = \mathrm{orth}(U_A)$), and standard LoRA without noise injection. All variants are evaluated using RoBERTa-base on SST-2, and the results are reported in Table 5. Our full method achieves the best adversarial performance while maintaining competitive clean accuracy. In contrast, random-subspace projection provides only marginal improvements over standard LoRA, suggesting that arbitrary noise injection is insufficient. These results demonstrate that the effectiveness of noise-based regularization depends on aligning the perturbations with the LoRA update subspace through $U$.

### 4.4 Robust Overfitting

Adversarial training can suffer from robust overfitting, where the model continues to fit the training objective while its adversarial generalization deteriorates (Rice et al., 2020). We examine this behavior by extending SST-2 training with RoBERTa-base and comparing ER-LoRA with representative baselines. As shown in Figure 5, the baselines show the typical pattern of robust overfitting: training error continues to decrease, while test adversarial error stops improving and begins to drift upward after the learning-rate drop.

ER-LoRA is less affected by this pattern. Its adversarial validation error remains comparatively stable during extended training, even as the training error continues to decrease. This suggests that the proposed regularizer improves robustness without encouraging the model to memorize adversarial artifacts from the training trajectory. We attribute this to the expectation-based closed-form regularizer: instead of repeatedly optimizing against a small set of hard perturbations, ER-LoRA smooths predictions over a local LoRA-aware perturbation distribution. This averaged robustness signal helps reduce the gap between training robustness and validation robustness.

## 5 Conclusion and Future Work

We have proposed ER-LoRA, an expectation-based regularizer that improves prediction stability in the LoRA update space. ER-LoRA uses a closed-form approximation to avoid adversarial example generation and expensive sampling while preserving the efficiency and clean accuracy of LoRA. Experiments show improved adversarial robustness with a favorable accuracy-efficiency trade-off. In future, we plan to extend ER-LoRA to instruction-tuned models and explore adaptive subspace construction for more diverse robustness settings.

## Limitations

This work studies adversarial robustness for LoRA-based fine-tuning mainly on English classification benchmarks. Although we evaluate ER-LoRA on multiple AdvGLUE tasks and both encoder- and decoder-only backbones, we do not cover multilingual settings, generation tasks, or very large-scale LLMs.

ER-LoRA also relies on a local Gaussian approximation in the LoRA update space. While this gives an efficient closed-form regularizer, it may not capture all non-local or highly discrete textual perturbations. Most experiments are run under a fixed training setup rather than extensive multi-seed evaluation due to computational constraints.

## References


Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021a. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

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

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Aneesh Sreevallabh Chivukula, Xinghao Yang, Wei Liu, Tianqing Zhu, and Wanlei Zhou. 2021. Game theoretical adversarial deep learning with variational adversaries. *IEEE Transactions on Knowledge and Data Engineering*, 33(11):3568–3581.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Junhao Dong, Seyed-Mohsen Moosavi-Dezfooli, Jianhuang Lai, and Xiaohua Xie. 2023. The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24678–24687.

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 (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. 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.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.

Google DeepMind. 2026. Gemma 4: Our most intelligent open models. Model card.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

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 *AAAI*.

Chunyuan Li, Hadi Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*.

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.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8410–8418.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32100–32121. PMLR.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*.

Donald W Marquardt. 1963. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441.

James Martens. 2010. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning*, pages 735–742.

James Martens. 2020. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*.

Preetum Nakkiran. 2019. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*.

Mingze Ni, Zhensu Sun, and Wei Liu. 2024. Reversible jump attack to textual classifiers with modification reduction. *Machine Learning*, 113(9):5907–5937.

Lin Pan, Chung-Wei Hang, Avirup Sil, and Saloni Potdar. 2022. Improved text classification via contrastive adversarial training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11130–11138.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Leslie Rice, Eric Wong, and Zico Kolter. 2020. Overfitting in adversarially robust deep learning. In *International conference on machine learning*, pages 8093–8104. PMLR.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019. Evaluating adversarial attacks against multiple fact verification systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2944–2953, Hong Kong, China. Association for Computational Linguistics.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*.

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. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6134–6150, Online. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. SemAttack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 176–205, Seattle, United States. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *Advances in Neural Information Processing Systems*.

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, Volume 1 (Long Papers)*, pages 1112–1122.

Zhizhou Yin, Fei Wang, Wei Liu, and Sanjay Chawla. 2018. Sparse feature attacks in adversarial learning. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1164–1177.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael I. Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7472–7482.

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*.

Hai Zhu, Qingyang Zhao, Weiwei Shang, Yuren Wu, and Kai Liu. 2024. Limeattack: Local explainable method for textual hard-label adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19759–19767.

# Appendix

## A Proofs

### A.1 Proof of Theorem 1

*Proof.* Using the Taylor approximation,

$$F(h+Uz) \approx F(h) + g^\top Uz + \frac{1}{2} z^\top U^\top H U z.$$

Substituting this into $\Phi(z;h) = F(h+Uz) - \frac{\tau}{2}\|z\|_2^2$ gives

$$\Phi(z;h) \approx F(h) + b^\top z - \frac{1}{2} z^\top \Lambda z,$$

where $b = U^\top g$ and $\Lambda = \tau I_r - U^\top H U$. Since $F(h)$ is independent of $z$, it is absorbed into the normalizing constant:

$$q(z \mid h) \propto \exp\left( b^\top z - \frac{1}{2} z^\top \Lambda z \right).$$

Completing the square,

$$b^\top z - \frac{1}{2} z^\top \Lambda z = -\frac{1}{2}(z-\mu)^\top \Lambda (z-\mu) + \frac{1}{2}\mu^\top \Lambda \mu,$$

where $\mu = \Lambda^{-1} b = \Lambda^{-1} U^\top g$. The remaining term $\frac{1}{2}\mu^\top \Lambda \mu$ is also independent of $z$ and is absorbed into the normalizing constant. Since $\Lambda \succ 0$, this Gaussian density is normalizable, with mean $\mu$ and covariance $\Sigma = \Lambda^{-1}$. □

### A.2 Proof of Corollary 1

*Proof.* At $z = 0$, the clean and perturbed predictive distributions coincide, so the KL divergence is zero. Its first-order derivative with respect to $z$ also vanishes at $z = 0$, since the KL divergence is locally minimized when the two distributions are equal. Therefore, under the second-order approximation around $z = 0$, we have

$$D_{\mathrm{KL}}(p_\theta(h) \,\|\, p_\theta(h+Uz)) \approx \frac{1}{2} z^\top A_\theta z. \tag{23}$$

Taking expectation over $z \sim \mathcal{N}(\mu, \Sigma)$ gives

$$R_{\mathrm{ER}}(\theta; h) \approx \frac{1}{2} \mathbb{E}[z^\top A_\theta z]. \tag{24}$$

Since $z^\top A_\theta z = \mathrm{tr}(A_\theta z z^\top)$ and $\mathbb{E}[z z^\top] = \Sigma + \mu\mu^\top$, we obtain

$$R_{\mathrm{ER}}(\theta; h) \approx \frac{1}{2} \mathrm{tr}\Big( A_\theta (\Sigma + \mu\mu^\top) \Big) \tag{25}$$

$$= \frac{1}{2} \mathrm{tr}(A_\theta \Sigma) + \frac{1}{2} \mu^\top A_\theta \mu. \tag{26}$$

This proves Eq. (17). □

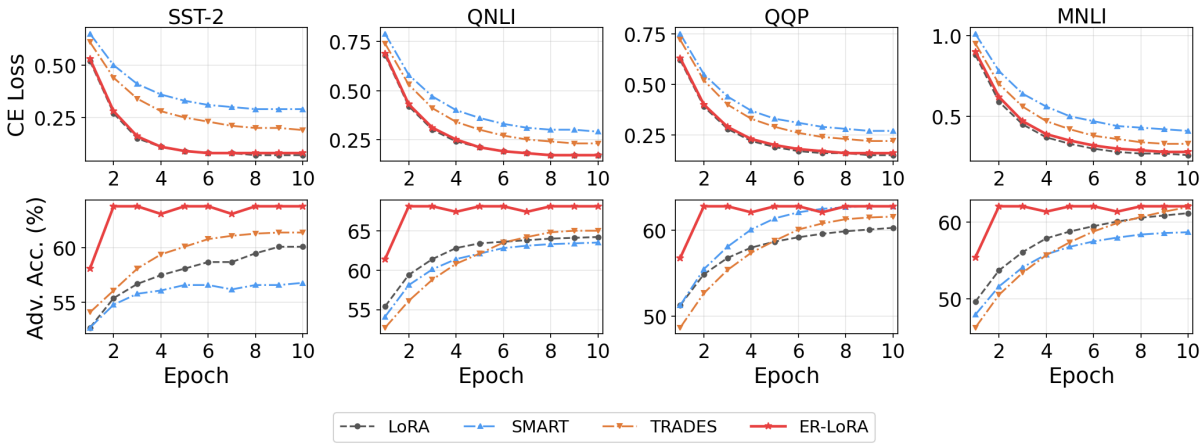



Figure 6: Training curves on DeBERTa-base across four AdvGLUE tasks. Each panel reports cross-entropy loss and adversarial accuracy over training epochs. ER-LoRA maintains strong adversarial accuracy throughout training while showing comparable clean-loss convergence to the baselines.

## B Additional Experimental Details

**Datasets and metrics.** We evaluate ER-LoRA on four AdvGLUE tasks: AdvSST-2 (Socher et al., 2013), AdvQQP (Wang et al., 2018), AdvMNLI (Williams et al., 2018), and AdvQNLI (Demszky et al., 2018). For each task, clean accuracy is measured on the original development split, while adversarial accuracy is measured on the corresponding AdvGLUE split. We report both task-level results and averages across tasks so that robustness gains can be compared together with clean-task retention.

**Backbones.** The main encoder experiments use RoBERTa-base (Liu et al., 2019) and DeBERTa-base (He et al., 2021). These two backbones allow us to test whether the proposed regularizer is stable across different pretrained encoder families. To further examine architectural transfer, we also evaluate compact decoder-only language models, including Gemma-4-E2B (Google DeepMind, 2026) and Qwen3.5-0.8B (Qwen Team, 2026).

**Baselines.** We compare ER-LoRA with standard LoRA (Hu et al., 2022), LoRA-r32, FreeLB-LoRA (Zhu et al., 2020), SMART-LoRA (Jiang et al., 2020), TRADES-LoRA, UIAT-LoRA (Dong et al., 2023), and PWWS-AT-LoRA (Ren et al., 2019). LoRA-r32 increases the LoRA rank to $r = 32$ and is included to test whether robustness can be improved simply by increasing adapter capacity. FreeLB-LoRA, SMART-LoRA, TRADES-LoRA, and UIAT-LoRA adapt robust training objectives to LoRA fine-tuning. PWWS-AT-LoRA instead uses offline adversarial examples generated by PWWS and mixes them with clean training data.

**Implementation details.** All methods are trained and evaluated under the same protocol for a fair comparison. For a given backbone and task, we use the same data split, tokenizer, optimizer, batch size, learning-rate schedule, and evaluation procedure across methods. Unless otherwise specified, the LoRA rank is set to $r = 8$, and LoRA modules are applied to attention projection matrices. The regularization weight of ER-LoRA is selected on the validation set and then fixed for the reported evaluation. For the hyperparameter $\lambda$, we hold out 5% of the provided adversarial set as the validation set and select $\lambda$ from $\{0.1, 0.3, 0.5, 1, 3, 5, 10\}$ based on validation performance. When a pre-generated adversarial set is unavailable, the validation set can instead be constructed using dynamic attack methods, such as PSO (Zang et al., 2020), TextFooler (TF) (Jin et al., 2020), and T3 (Wang et al., 2020).

## C Training Curves on DeBERTa-base

To complement the RoBERTa-base curves in the main text, we report the full DeBERTa-base training dynamics in Figure 6. The curves track both cross-entropy loss and adversarial accuracy across all four AdvGLUE tasks. They show that the behavior observed in the main text is not specific to RoBERTa-base: ER-LoRA reaches competitive clean loss convergence while maintaining stronger and more stable adversarial accuracy during training. This supports the conclusion that the expectation-based regularizer provides a consistent robustness signal across encoder backbones.

## D Code for the main experiments

We have released the implementation, training scripts, and evaluation configuration files for the main experiments. The release includes the ER-LoRA objective, baseline configurations, dataset preprocessing scripts, and commands for reproducing the AdvGLUE and decoder-only LLM experiments. The code is available at the following link: https://github.com/MingzeLucasNi/ER-LoRA.git

## E Decoder-only Evaluation Protocol

For decoder-only models, classification is formulated as next-token prediction with fixed prompts and single-token verbalizers:

- **SST-2**: `Review: {sentence} Sentiment:` (positive/negative)
- **QQP**: `Question 1: {q1} Question 2: {q2} Same? Answer:` (Yes/No)
- **QNLI**: `Question: {question} Sentence: {sentence} Answered? Answer:` (Yes/No)
- **MNLI**: `Premise: {premise} Hypothesis: {hypothesis} Entailed? Answer:` (Yes/Maybe/No)