

# ADVERSARIALLY PRETRAINED TRANSFORMERS MAY BE UNIVERSALLY ROBUST IN-CONTEXT LEARNERS

**Soichiro Kumano**

The University of Tokyo  
kumano@cvm.t.u-tokyo.ac.jp

**Hiroshi Kera**

Chiba University, National Institute of Informatics  
kera@chiba-u.jp

**Toshihiko Yamasaki**

The University of Tokyo  
yamasaki@cvm.t.u-tokyo.ac.jp

## ABSTRACT

Adversarial training is one of the most effective defenses against adversarial attacks, but it incurs a high computational cost. In this study, we present the first theoretical analysis suggesting that adversarially pretrained transformers can serve as *universally robust foundation models*—models that can adapt robustly to diverse downstream tasks with only lightweight tuning. Specifically, we demonstrate that single-layer linear transformers, after adversarial pretraining across a variety of classification tasks, can generalize robustly to unseen classification tasks through in-context learning from clean demonstrations (i.e., without requiring additional adversarial training or examples). This universal robustness stems from the model’s ability to adaptively focus on robust features within given tasks. We also identify two open challenges for attaining robustness: the accuracy–robustness trade-off and sample-hungry training. This study initiates the discussion on the utility of universally robust foundation models. While their training is expensive, the investment would prove worthwhile as downstream tasks can obtain adversarial robustness for free. The code is available at <https://github.com/s-kumano/universally-robust-in-context-learner>.

## 1 INTRODUCTION

Adversarial examples—subtle and often imperceptible perturbations to inputs that lead machine learning models to make incorrect predictions—reveal a fundamental vulnerability in modern deep learning systems (Szegedy et al., 2014). Adversarial training is one of the most effective defenses against such attacks (Goodfellow et al., 2015; Madry et al., 2018), where classification loss is minimized under worst-case (i.e., adversarial) perturbations. This min–max optimization significantly increases the computational cost compared to standard training. Despite extensive efforts to develop alternative defenses, most such approaches have subsequently been shown to offer only spurious robustness (Athalye et al., 2018; Croce & Hein, 2020; Tramer et al., 2020). Consequently, adversarial training remains the de facto standard, and practitioners must incur this cost to obtain adversarially robust models.

Recently, it has become standard practice to leverage foundation models for target tasks. Thanks to large-scale pretraining, these models can be adapted to diverse downstream tasks through lightweight tuning. This raises a natural question: *Can adversarially trained foundation models enable efficient and robust adaptation to a wide range of downstream tasks?* Although training such models is expensive, the investment would be justified if numerous downstream tasks could inherit adversarial robustness for free, without requiring costly adversarial training themselves. While this is a promising research direction, the utility of such *universally robust foundation models* remains largely unexplored, as the computational and financial costs make empirical evaluation across multiple runs impractical.

In this study, we present the first theoretical analysis suggesting that adversarially pretrained transformers can serve as universally robust foundation models. Specifically, we show that single-layer linear transformers, after adversarial pretraining across a variety of classification tasks, can generalize

robustly to previously unseen classification tasks through in-context learning (Brown et al., 2020) from clean demonstrations. Namely, these transformers achieve robust adaptation without requiring additional adversarial examples or training. In-context learning is a transformer capability that enables efficient adaptation to new tasks from a few input–output demonstrations in the prompt, without any parameter updates.

Our analysis builds upon the conceptual framework of robust features (class-discriminative and human-interpretable) and non-robust features (human-imperceptible yet predictive) (Ilyas et al., 2019; Tsipras et al., 2019). Based on this framework, we show that adversarially pretrained single-layer linear transformers can adaptively focus on robust features within each downstream task, rather than non-robust or non-predictive features, thereby achieving universal robustness. This framework also reveals that universal robustness holds under mild conditions, except in an unrealistic scenario where non-robust features overwhelmingly outnumber robust ones.

We also show that two open challenges in robust machine learning (Schmidt et al., 2018; Tsipras et al., 2019) persist in our setting. First, adversarially pretrained single-layer linear transformers exhibit lower clean accuracy than their standardly pretrained counterparts. Second, to achieve clean accuracy comparable to standard models, these transformers require more in-context demonstrations.

Our contributions are summarized as follows:

- We provide the first theoretical evidence for universally robust foundation models: under mild conditions, adversarially pretrained transformers with a single linear self-attention layer can adapt robustly to unseen classification tasks through in-context learning.
- Based on the framework of robust and non-robust features, we characterize the condition for successful robust adaptation. Moreover, we show that universal robustness arises from the models’ adaptive focus on robust features within each downstream task.
- We identify two open problems for these transformers: the accuracy–robustness trade-off and sample-hungry in-context learning.

This study explores the potential of universally robust foundation models, which can provide diverse downstream tasks with adversarial robustness without additional adversarial training. A key challenge is the cost of adversarial pretraining. We assume that, as with standard foundation models, such efforts would be undertaken by large organizations, which could offset development costs through licensing or API fees. The growing demand for safe and reliable AI strengthens this incentive. Encouragingly, advances in acceleration techniques for adversarial training, such as fast adversarial training (Wong et al., 2020) and adversarial finetuning (Jeddi et al., 2020), suggest that the cost of adversarial training could approach the cost of standard training. We view our theoretical analysis as an important first step toward fostering the practical development of universally robust foundation models.

## 2 RELATED WORK

Additional related work can be found in [Appendix A](#).

**Adversarial Training.** Adversarial training (Goodfellow et al., 2015; Madry et al., 2018), which augments training data with adversarial examples (Szegedy et al., 2014), is one of the most effective adversarial defenses. Its major limitation is the high computational cost. To address this, several methods have focused on efficient generation of adversarial examples (Andriushchenko & Flammarion, 2020; Kim et al., 2021; Park & Lee, 2021; Shafahi et al., 2019; Wong et al., 2020; Zhang et al., 2019a) and adversarial finetuning (Jeddi et al., 2020; Mao et al., 2023; Suzuki et al., 2023; Wang et al., 2024a). However, these methods require task-specific adversarial training. In this study, we introduce the concept of universally robust foundation models, which can adapt robustly to a wide range of downstream tasks without requiring any adversarial training or examples.

**Robust and Non-Robust Features.** It is widely hypothesized that adversarial vulnerability arises from models’ reliance on non-robust features (Ilyas et al., 2019; Tsipras et al., 2019). While robust features are class-discriminative, human-interpretable, and semantically meaningful, non-robust features are subtle, often imperceptible to humans, yet statistically correlated with labels and therefore predictive. Humans can rely only on robust features, whereas models can leverage both types of features to maximize accuracy. Tsipras et al. (2019) showed that standard classifiers

depend heavily on non-robust features, making them vulnerable to adversarial perturbations that can manipulate these subtle features. They also showed that adversarial training encourages models to rely primarily on robust features, which enhances robustness but often reduces clean accuracy—a phenomenon known as the accuracy–robustness trade-off (Dobriban et al., 2023; Mehrabi et al., 2021; Raghunathan et al., 2019; 2020; Su et al., 2018; Tsipras et al., 2019; Yang et al., 2020; Zhang et al., 2019b). Subsequent studies have confirmed that adversarially trained neural networks exhibit a greater reliance on robust features (Augustin et al., 2020; Chalasani et al., 2020; Engstrom et al., 2019; Etmann et al., 2019; Kaur et al., 2019; Santurkar et al., 2019; Srinivas et al., 2023; Tsipras et al., 2019; Zhang & Zhu, 2019). In this study, we incorporate the concept of robust and non-robust features into the data assumptions for our theoretical analysis. Based on this framework, we find that adversarially pretrained single-layer linear transformers prioritize robust features over non-robust features, and exhibit the accuracy–robustness trade-off.

### 3 THEORETICAL RESULTS

**Notation.** For  $n \in \mathbb{N}$ , let  $[n] := \{1, \dots, n\}$ . Denote the  $i$ -th element of a vector  $\mathbf{a}$  by  $a_i$ , and the element in the  $i$ -th row and  $j$ -th column of a matrix  $\mathbf{A}$  by  $A_{i,j}$ . Let  $U(\mathcal{S})$  be the uniform distribution over a set  $\mathcal{S}$ . The sign function is denoted as  $\text{sgn}(\cdot)$ . For  $d_1, d_2 \in \mathbb{N}$ , let  $\mathbf{1}_{d_1}$  and  $\mathbf{1}_{d_1, d_2}$  be the  $d_1$ -dimensional all-ones vector and  $d_1 \times d_2$  all-ones matrix, respectively. The  $d_1 \times d_1$  identity matrix is denoted as  $\mathbf{I}_{d_1}$ . Similarly, we write the all-zeros vector and matrix as  $\mathbf{0}_{d_1}$  and  $\mathbf{0}_{d_1, d_2}$ , respectively. We use  $\gtrsim, \lesssim$ , and  $\approx$  only to hide constant factors in informal statements.

#### 3.1 PROBLEM SETUP

**Overview.** We adversarially train a single-layer linear transformer on  $d \in \mathbb{N}$  distinct datasets. The  $c$ -th training data distribution is denoted by  $\mathcal{D}_c^{\text{tr}}$  for  $c \in [d]$ . The  $c$ -th dataset consists of  $N+1$  samples,  $\{(\mathbf{x}_n^{(c)}, y_n^{(c)})\}_{n=1}^{N+1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}_c^{\text{tr}}$ . The transformer is encouraged to adaptively learn data structures from  $N$  clean in-context demonstrations  $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$  and to generalize to the  $(N+1)$ -th perturbed sample  $\mathbf{x}_{N+1} + \mathbf{\Delta}$ , where  $\mathbf{\Delta}$  represents an adversarial perturbation. We then evaluate the adversarial robustness of the trained transformer on a test dataset  $\{(\mathbf{x}_n^{\text{te}}, y_n^{\text{te}})\}_{n=1}^{N+1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}^{\text{te}}$ , which may exhibit different structures from any training distributions.

**Transformer.** We first define the input sequence for a transformer as

$$\mathbf{Z}_{\mathbf{\Delta}} := \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N & \mathbf{x}_{N+1} + \mathbf{\Delta} \\ y_1 & y_2 & \cdots & y_N & 0 \end{bmatrix} \in \mathbb{R}^{(d+1) \times (N+1)}, \quad (1)$$

where  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^d$  are training data,  $y_1, \dots, y_N \in \{\pm 1\}$  are their binary labels,  $\mathbf{x}_{N+1} \in \mathbb{R}^d$  is a test sample (query), and  $\mathbf{\Delta} \in \mathbb{R}^d$  is an adversarial perturbation (see below). The transformer is expected to learn data structures adaptively from  $N$  demonstrations  $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$  and to predict the label of  $\mathbf{x}_{N+1}$ . The  $(d+1, N+1)$ -th element of  $\mathbf{Z}_{\mathbf{\Delta}}$  serves as a placeholder for the prediction of  $\mathbf{x}_{N+1} + \mathbf{\Delta}$ . We define a single-layer linear transformer  $\mathbf{f} : \mathbb{R}^{(d+1) \times (N+1)} \rightarrow \mathbb{R}^{(d+1) \times (N+1)}$ , which is commonly employed in theoretical studies of in-context learning (Ahn et al., 2023; Cheng et al., 2024; Gatmiry et al., 2024; Mahankali et al., 2024; Zhang et al., 2024b), as follows:

$$\mathbf{f}(\mathbf{Z}_{\mathbf{\Delta}}; \mathbf{P}, \mathbf{Q}) := \frac{1}{N} \mathbf{P} \mathbf{Z}_{\mathbf{\Delta}} \mathbf{M} \mathbf{Z}_{\mathbf{\Delta}}^{\top} \mathbf{Q} \mathbf{Z}_{\mathbf{\Delta}}, \quad \mathbf{M} := \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(N+1) \times (N+1)}, \quad (2)$$

where  $\mathbf{P} \in \mathbb{R}^{(d+1) \times (d+1)}$  serves as the value weight matrix and  $\mathbf{Q} \in \mathbb{R}^{(d+1) \times (d+1)}$  serves as the product of the key and query weight matrices. Following prior work on in-context learning (Ahn et al., 2023; Cheng et al., 2024; Gatmiry et al., 2024; Li et al., 2025), we adopt a mask matrix  $\mathbf{M}$  to prevent the tokens from attending to the query token.

**Training Data Distribution.** The transformer is pretrained on  $d$  distinct datasets. Inspired by Tsipras et al. (2019), we consider the following data structure that explicitly separates robust and non-robust features (cf. Section 2) according to their dimensional indices:

**Assumption 3.1** (Individual training data distribution). Let  $c \in [d]$  be the index of a training data distribution and  $\mathcal{D}_c^{\text{tr}}$  be the  $c$ -th distribution. A sample  $(\mathbf{x}, y) \sim \mathcal{D}_c^{\text{tr}}$  satisfies the following:

$$y \sim U(\{\pm 1\}), \quad x_c = y, \quad \forall i \in [d], i \neq c: x_i \sim \begin{cases} U([0, y\lambda]) & (y = 1) \\ U([y\lambda, 0]) & (y = -1) \end{cases}, \quad (3)$$

where  $0 < \lambda < 1$ . For any  $i \neq j$ ,  $x_i$  and  $x_j$  are independent, given  $y$ .

In this distribution, each sample has a feature that is strongly correlated with its label (i.e., a robust feature) in the  $c$ -th dimension and weakly correlated features (i.e., non-robust features) in the remaining dimensions. The correlation between the non-robust features and the label is bounded by  $\lambda$ . The robust features mimic human-interpretable, semantically meaningful attributes in natural objects (e.g., shape). The non-robust features mimic human-imperceptible yet predictive attributes (e.g., texture).

**Test Data Distribution.** The test data distribution may exhibit more diverse structures than the training data distributions, and may include non-predictive features in addition to robust and non-robust features.

**Assumption 3.2** (Test data distribution). Let the index sets of robust, non-robust, and irrelevant features be  $\mathcal{S}_{\text{rob}}, \mathcal{S}_{\text{vul}}, \mathcal{S}_{\text{irr}} \subset [d]$ , respectively. Suppose that these sets are disjoint, i.e.,  $\mathcal{S}_{\text{rob}} \cap \mathcal{S}_{\text{vul}} = \mathcal{S}_{\text{vul}} \cap \mathcal{S}_{\text{irr}} = \mathcal{S}_{\text{irr}} \cap \mathcal{S}_{\text{rob}} = \emptyset$  and that  $\mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}} \cup \mathcal{S}_{\text{irr}} = [d]$ . Let the number of robust, non-robust, and irrelevant features be  $d_{\text{rob}} := |\mathcal{S}_{\text{rob}}|$ ,  $d_{\text{vul}} := |\mathcal{S}_{\text{vul}}|$ , and  $d_{\text{irr}} := |\mathcal{S}_{\text{irr}}|$ , respectively. Let the scales of the robust, non-robust, and irrelevant features be  $\alpha > 0$ ,  $\beta > 0$ , and  $\gamma \geq 0$ , respectively. Let  $\mathcal{D}^{\text{te}}$  be a test data distribution. A sample  $(\mathbf{x}, y) \sim \mathcal{D}^{\text{te}}$  satisfies the following:

(1. Label) The label  $y$  follows the uniform distribution  $U(\{\pm 1\})$ .

(2. Expectation and Moments) For every  $i \in \mathcal{S}_{\text{irr}}$ ,  $\mathbb{E}[x_i] = 0$ . For every  $i \in [d]$  and  $n \in \{2, 3, 4\}$ , there exist constants  $C_i > 0$  and  $C_{i,n} \geq 0$  such that

$$\mathbb{E}[yx_i] = \begin{cases} C_i \alpha & (i \in \mathcal{S}_{\text{rob}}) \\ C_i \beta & (i \in \mathcal{S}_{\text{vul}}) \\ 0 & (i \in \mathcal{S}_{\text{irr}}) \end{cases}, \quad |\mathbb{E}[(yx_i - \mathbb{E}[yx_i])^n]| \leq \begin{cases} C_{i,n} \alpha^n & (i \in \mathcal{S}_{\text{rob}}) \\ C_{i,n} \beta^n & (i \in \mathcal{S}_{\text{vul}}) \\ C_{i,n} \gamma^n & (i \in \mathcal{S}_{\text{irr}}) \end{cases}. \quad (4)$$

(3. Covariance) There exist constants  $0 \leq q_{\text{rob}}, q_{\text{vul}} < 1$  such that

$$\left\{ \begin{array}{l} i \in \mathcal{S}_{\text{rob}} \\ \sum_{j \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} \mathbb{E}[(yx_i - \mathbb{E}[yx_i])(yx_j - \mathbb{E}[yx_j])] < 0 \end{array} \right\} \leq q_{\text{rob}} d_{\text{rob}}, \quad (5)$$

$$\left\{ \begin{array}{l} i \in \mathcal{S}_{\text{vul}} \\ \sum_{j \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} \mathbb{E}[(yx_i - \mathbb{E}[yx_i])(yx_j - \mathbb{E}[yx_j])] < 0 \end{array} \right\} \leq q_{\text{vul}} d_{\text{vul}}. \quad (6)$$

(4. Independence) For every  $i \in \mathcal{S}_{\text{irr}}$ ,  $x_i$  is independent of  $y$  and all  $x_j$  for  $j \neq i$ .

In contrast to the training distributions, the test distribution may contain  $d_{\text{rob}}$  robust features and  $d_{\text{irr}}$  irrelevant features. These irrelevant features simulate natural noise or redundant dimensions commonly found in real-world data. For example, in MNIST (Deng, 2012), the top-left pixel is always zero and thus not predictive. Assumption 4 requires each irrelevant feature to be independent of both the label and all other features. The robust and non-robust features are not necessarily mutually independent.

Assumption 2 (Expectation) ensures that the robust and non-robust features are positively correlated with the label. Given sufficient data, it is always possible to preprocess features to positively align with their binary labels. For example, with a large  $N$ , this can be achieved by multiplying each feature  $x_i$  by  $\text{sgn}(\mathbb{E}[yx_i]) \approx \text{sgn}(\sum_{n=1}^N y_n x_{n,i})$ , ensuring  $\mathbb{E}[y \text{sgn}(\mathbb{E}[yx_i]) x_i] = |\mathbb{E}[yx_i]| \geq 0$ .

Assumption 2 (Moments) bounds the  $n$ -th central moment of each feature by a constant multiple of the  $n$ -th power of its scale for  $n \in \{2, 3, 4\}$ . This condition ensures that the feature distribution does not exhibit excessively large fluctuations relative to its scale ( $n = 2$ ), extreme asymmetry ( $n = 3$ ), or heavy tails ( $n = 4$ ). For example, with appropriate constant factors, exponential distributions satisfy this condition. Moreover, empirical studies suggest that pixel values (or contrasts) after typical filtering (e.g., Gabor filtering) approximately follow exponential distributions (Ruderman, 1994;

Srivastava et al., 2003). This observation suggests that filtered pixel values (and contrasts) are broadly consistent with this assumption.

Assumption 3 bounds the number of features whose total covariance with other informative features (i.e., robust and non-robust features) is negative. As stated in [Theorem 3.6](#), we typically assume that  $q_{\text{rob}}$  and  $q_{\text{vul}}$  are small (but not necessarily infinitesimal). This assumption prevents unrealistic cases where useful features are overly anti-correlated with others, which could hinder learning. When all the predictive features are conditionally independent given the label,  $q_{\text{rob}} = 0$  and  $q_{\text{vul}} = 0$  satisfy this assumption. Empirically,  $q_{\text{rob}}$  and  $q_{\text{vul}}$  appear to be small in real-world datasets (cf. [Fig. A2](#)).

These conditions encompass a wide class of realistic data distributions.

- **Example 1: Training data distribution.** Each training distribution  $\mathcal{D}_c^{\text{tr}}$  is a special case of the test distribution  $\mathcal{D}^{\text{te}}$ . Specifically, it contains  $d_{\text{rob}} = 1$  robust feature with scale  $\alpha \approx 1$  and  $d_{\text{vul}} = d - 1$  non-robust features with scale  $\beta \approx \lambda$ . There are no irrelevant features, i.e.,  $d_{\text{irr}} = 0$ . By construction and due to the properties of uniform distribution, this distribution satisfies all the conditions in [Assumption 3.2](#).
- **Example 2: Standard distributions.** With appropriate constant factors, the test distribution class includes standard distributions, such as uniform, normal, exponential, beta, gamma, Bernoulli, binomial, etc. For example, consider the normal distribution. For  $i \in \mathcal{S}_{\text{rob}}$ , Assumption 2 is satisfied by setting  $yx_i \sim \mathcal{N}(\alpha, \alpha^2)$  with  $C_i = 1$ ,  $C_{i,2} = 1$ ,  $C_{i,3} = 0$ , and  $C_{i,4} = 3$ . Assumptions 3 and 4 are satisfied when all the features are mutually independent.
- **Example 3: MNIST/Fashion-MNIST/CIFAR-10.** Empirical evidence suggests that preprocessed MNIST ([Deng, 2012](#)), Fashion-MNIST ([Xiao et al., 2017](#)), and CIFAR-10 ([Krizhevsky, 2009](#)) approximately satisfy our assumptions. Consider MNIST. Let  $\{\mathbf{x}_n^{(0)}\}_{n=1}^N, \{\mathbf{x}_n^{(1)}\}_{n=1}^N \in [0, 1]^{784}$  denote the samples of digits zero and one, respectively. We assign  $y = 1$  to digit zero and  $y = -1$  to digit one. We center the data via  $\mathbf{x}' \leftarrow \mathbf{x} - \bar{\mathbf{x}}$  with  $\bar{\mathbf{x}} := (1/2N) \sum_{n=1}^N (\mathbf{x}_n^{(0)} + \mathbf{x}_n^{(1)})$  and align features with the label using  $\mathbf{x}'' \leftarrow \text{sgn}(\sum_{n=1}^N (\mathbf{x}_n^{(0)} - \mathbf{x}_n^{(1)})) \odot \mathbf{x}'$ . In this representation, common background pixels have near-zero expectations (i.e.,  $\gamma \approx 0$ ), while discriminative pixels—such as the left and right arcs of zero or the vertical stroke of one—correlate strongly with the label (i.e.,  $\alpha \approx 0.2$ ) (cf. [Fig. A2](#)). Additionally, some pixels that are occasionally activated by atypical samples (e.g., corners activated by slanted digits) exhibit weak correlation with the label (i.e.,  $\beta \approx 0.01$ ), reflecting non-robust but predictive attributes. Empirical analysis reveals that most pixels exhibit positive total covariance with others, consistent with Assumption 3 (cf. [Fig. A2](#)). The main departure from [Assumption 3.2](#) is that real-world datasets exhibit a gradual transition in feature robustness rather than an explicit binary separation between robust and non-robust features.
- **Example 4: Linear combination of orthonormal bases.** Under mild conditions, any distribution in which robust and non-robust directions form an orthonormal basis can be transformed into our setting via principal component analysis (cf. [Appendix C](#)).

**Adversarial Attack.** We assume that the query (test sample)  $\mathbf{x}_{N+1}$  is subject to the adversarial perturbation  $\Delta$  constrained by the  $\ell_\infty$  norm, i.e.,  $\|\Delta\|_\infty \leq \epsilon$ , where  $\epsilon \geq 0$  denotes the perturbation budget. In practice,  $\epsilon$  is chosen to be consistent with the scale of non-robust features (e.g.,  $\epsilon \approx \lambda$  for the training distributions and  $\epsilon \approx \beta$  for the test distribution). This ensures that perturbations can manipulate non-robust features while leaving robust features intact and remaining imperceptible to humans.

**Pretraining With In-Context Loss.** For pretraining, we consider the following problem based on the in-context loss ([Ahn et al., 2023](#); [Bai et al., 2023](#); [Mahankali et al., 2024](#); [Zhang et al., 2024b](#)):

$$\min_{\mathbf{P}, \mathbf{Q} \in [0, 1]^{(d+1) \times (d+1)}} \mathbb{E}_{c \sim U([d]), \{\mathbf{x}_n, y_n\}_{n=1}^{N+1} \sim \mathcal{D}_c^{\text{tr}}} \left[ \max_{\|\Delta\|_\infty \leq \epsilon} -y_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}, \mathbf{Q})]_{d+1, N+1} \right]. \quad (7)$$

This formulation encourages the transformer to extract robust, generalizable representations from  $N$  clean in-context demonstrations and to accurately classify the adversarially perturbed query. We impose the constraint on the transformer parameters to prevent the problem from becoming ill-posed. We choose  $[0, 1]^d$  instead of  $[-1, 1]^d$  to simplify the theoretical derivation.

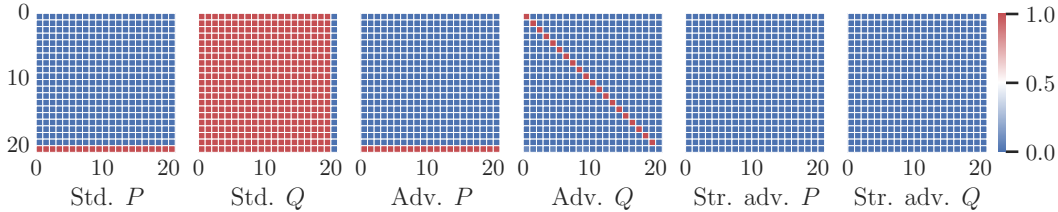


Figure 1: Parameter heatmaps learned via adversarial training (7) with  $d = 20$  and  $\lambda = 0.1$ . For the standard, adversarial, and strong adversarial regimes, we used  $\epsilon = 0$ ,  $\frac{1+(d-1)(\lambda/2)}{d} = 0.098$ , and  $\frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3} = 0.95$ , respectively. We optimized (7) by stochastic gradient descent. Detailed experimental settings can be found in Appendix D.

### 3.2 LINEAR CLASSIFIERS AND ORACLE

**Standard Linear Classifiers Extract All Features and Are Therefore Vulnerable.** As a warm-up, consider standard training of a linear classifier parameterized by  $\mathbf{w} \in \mathbb{R}^d$  on the  $c$ -th training distribution  $\mathcal{D}_c^{\text{tr}}$ . Standard training yields  $\mathbf{w}^{\text{std}} := \arg \min_{\mathbf{w} \in [0,1]^d} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_c^{\text{tr}}} [-y\mathbf{w}^\top \mathbf{x}] = \mathbf{1}_d$ . This classifier utilizes all the features, i.e., the robust feature at the  $c$ -th dimension and the other non-robust features. Although  $\mathbf{w}^{\text{std}}$  achieves correct predictions on clean samples ( $\mathbb{E}[y\mathbf{w}^{\text{std}\top} \mathbf{x}] > 0$ ), it is vulnerable to adversarial perturbations:  $\mathbb{E}[\min_{\|\Delta\|_\infty \leq \epsilon} y\mathbf{w}^{\text{std}\top} (\mathbf{x} + \Delta)] \leq 0$  for  $\epsilon \geq \frac{1+(d-1)(\lambda/2)}{d}$ <sup>1</sup>. This implies that when  $d$  is small, the perturbation must satisfy  $\epsilon \gtrsim 1$  to flip the prediction. In this regime, the perturbation alters the robust feature and is no longer human-imperceptible. However, as  $d$  increases, the threshold decreases to  $\epsilon \gtrsim \lambda$ , which matches the scale of the non-robust features. Such perturbations are human-imperceptible yet sufficient to cause misclassification.

**Linear Classifiers Can Be Specifically Robust but Not Universally Robust.** Consider adversarial training:  $\min_{\mathbf{w} \in [0,1]^d} \mathbb{E}[\max_{\|\Delta\|_\infty \leq \epsilon} -y\mathbf{w}^\top (\mathbf{x} + \Delta)]$ . For  $\frac{\lambda}{2} \leq \epsilon < 1$ , the optimal solution  $\mathbf{w}^{\text{adv}}$  equals one at the  $c$ -th dimension and zero otherwise. The classifier relies solely on the robust feature at the  $c$ -th dimension and ignores the other non-robust features. Unlike the standard model, this model can correctly classify both clean and adversarial samples for  $0 \leq \epsilon < 1$ ; thus, linear classifiers can be robust to a specific training distribution. However,  $\mathbf{w}^{\text{adv}}$  tailored to  $\mathcal{D}_c^{\text{tr}}$  is vulnerable on the other distributions  $\mathcal{D}_{c'}^{\text{tr}}$  indexed by  $c' \neq c$ ; thus, linear classifiers cannot be universally robust.

**Universally Robust Classifiers Exist.** Although linear classifiers cannot exhibit universal robustness across all  $c$ , universally robust classifiers do exist. For example, the classifier  $h(\mathbf{x}) := \text{sgn}(x_i)$  with  $i := \arg \max_{j \in [d]} |x_j|$  always produces correct predictions for clean data  $\mathbf{x} \sim \mathcal{D}_c^{\text{tr}}$  for any  $c$  and perturbed data  $\mathbf{x} + \Delta$  with  $\|\Delta\|_\infty \leq \frac{1}{2}$ .

### 3.3 ADVERSARIAL PRETRAINING

In this section, we analyze a global minimizer of the minimization problem (7).

**Optimization Challenges.** Although the training distributions are simple, the minimization problem (7) remains nontrivial due to the non-linearity and non-convexity of the model with respect to the trainable parameters  $\mathbf{P}$  and  $\mathbf{Q}$ . The non-linearity of the self-attention and the inner-maximization are also obstacles. Indeed, the minimization problem (7) can be reformulated as the following non-linear maximization problem:

**Lemma 3.3** (Transformation of original optimization problem). The minimization problem (7) can be transformed into the maximization problem  $\max_{\mathbf{b} \in \{0,1\}^{d+1}} \sum_{i=1}^{d(d+1)} \max(0, \sum_{j=1}^{d+1} b_j h_{i,j})$ , where  $h_{i,j} \in \mathbb{R}$  is a constant depending on  $(i, j)$ , and there exists a mapping from  $\mathbf{b}$  to  $\mathbf{P}$  and  $\mathbf{Q}$ .

The proof can be found in Appendix E. This lemma highlights the inherent difficulty of optimizing (7), as it requires selecting a binary vector  $\mathbf{b}$  that balances  $d(d+1)$  interdependent non-linear terms.

<sup>1</sup> $\mathbb{E}[\min_{\|\Delta\|_\infty \leq \epsilon} y\mathbf{w}^{\text{std}\top} (\mathbf{x} + \Delta)] = \mathbf{w}^{\text{std}\top} (\mathbb{E}[y\mathbf{x}] - \epsilon \mathbf{1}_d) = \{1 + (d-1)(\lambda/2)\} - d\epsilon \leq 0$ .

**Global Solution.** By exploiting the symmetric property of  $\mathbf{b}$  and further transformation of the problem in [Lemma 3.3](#), we identify the global solutions to (7) for certain perturbation regimes.

**Theorem 3.4** (Parameters learned via adversarial pretraining). The global minimizers of (7) are

$$\begin{aligned} (1. \text{ Standard; } \epsilon = 0) \quad & \mathbf{P} = \mathbf{P}^{\text{std}} := \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{1}_{d+1}^\top \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \mathbf{Q}^{\text{std}} := [\mathbf{1}_{d+1,d} \quad \mathbf{0}_{d+1}]. \\ (2. \text{ Adversarial; } \epsilon = \frac{1+(d-1)(\lambda/2)}{d}) \quad & \mathbf{P} = \mathbf{P}^{\text{adv}} := \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{1}_{d+1}^\top \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \mathbf{Q}^{\text{adv}} := \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix}. \\ (3. \text{ Strongly adversarial; } \epsilon \geq \frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3}) \quad & \mathbf{P} = \mathbf{0}_{d+1,d+1} \quad \text{and} \quad \mathbf{Q} = \mathbf{0}_{d+1,d+1}. \end{aligned}$$

The proof and optimal parameters for different  $\epsilon$  can be found in [Appendix E](#). Importantly, the optimal  $\mathbf{P}$  and  $\mathbf{Q}$  are independent of any specific training distribution (i.e., index  $c$ ), reflecting that the transformer acquires learning capability from demonstrations rather than memorizing individual tasks. Experimental results obtained using gradient descent align with our theoretical predictions ([Fig. 1](#)).

**Failure Case.** In the strongly adversarial regime, the global optimum becomes  $\mathbf{P} = \mathbf{Q} = \mathbf{0}$ , causing the transformer to always output zero regardless of the input. Namely, no universally robust single-layer linear transformers exist, despite the existence of universally robust classifiers (cf. [Section 3.2](#)). The perturbation scale  $\epsilon \geq \frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3}$  decreases in  $d$ : it transitions from  $\epsilon = 1$  when  $d = 1$  to  $\epsilon \rightarrow \frac{\lambda}{2}$  as  $d \rightarrow \infty$ . In moderate dimensions ( $d \approx \frac{1}{\lambda}$ ), adversarial perturbations must satisfy  $\epsilon \gtrsim 1$  to break the robustness. They are comparable to the scale of the robust features and thus perceptible to humans, contradicting the concept of adversarial perturbations. However, in extremely high dimensions ( $d \gtrsim \frac{1}{\lambda^2}$ ), it suffices to perturb by only  $\epsilon \gtrsim \lambda$ , which is on the same scale as the non-robust features and is typically imperceptible, thus preserving the concept of adversarial perturbations. This can be rephrased as follows: under our training distributions, single-layer linear transformers cannot achieve universal robustness when the non-robust dimensions (i.e.,  $d - 1$ ) substantially outnumber the single robust dimension.

### 3.4 UNIVERSAL ROBUSTNESS

In this section, we show that adversarial pretraining, combined with in-context learning from clean demonstrations, can yield universal robustness on both seen and unseen distributions.

**Standard Pretraining Leads to Vulnerability.** We begin by showing that the standard model fails to classify adversarially perturbed inputs.

**Theorem 3.5** (Standard pretraining case). There exist a constant  $C > 0$  and a strictly positive function  $g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma)$  such that

$$\begin{aligned} & \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}^{\text{te}}} \left[ \min_{\|\Delta\|_\infty \leq \epsilon} y_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}^{\text{std}}, \mathbf{Q}^{\text{std}})]_{d+1, N+1} \right] \\ & \leq g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) \left\{ \underbrace{C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta)}_{\text{Prediction for original data}} - \underbrace{(d_{\text{rob}} + d_{\text{vul}} + d_{\text{irr}})\epsilon}_{\text{Adversarial effect}} \right\}. \quad (8) \end{aligned}$$

The proof can be found in [Appendix F](#). This result analyzes the expectation of the product of the true label and the model prediction for the query. A positive value indicates correct classification, whereas a nonpositive value indicates misclassification. Since  $g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma)$  is always positive, when  $C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta) - (d_{\text{rob}} + d_{\text{vul}} + d_{\text{irr}})\epsilon$  is nonpositive, this implies incorrect classification.

*Standard models extract both robust and non-robust features and thus are vulnerable.* Assume  $d_{\text{irr}} = 0$ . Like standard linear classifiers, the standard model leverages both robust features  $d_{\text{rob}}\alpha$  and non-robust features  $d_{\text{vul}}\beta$ . This also leads to vulnerability to adversarial perturbations,  $-(d_{\text{rob}} + d_{\text{vul}})\epsilon$ . The prediction becomes incorrect when  $C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta) - (d_{\text{rob}} + d_{\text{vul}})\epsilon \leq 0$ , i.e., when  $\epsilon \gtrsim \frac{d_{\text{rob}}\alpha + d_{\text{vul}}\beta}{d_{\text{rob}} + d_{\text{vul}}}$ . When the perturbation size  $\epsilon$  is on the same scale as the non-robust features ( $\epsilon \lesssim \beta$ ),

the inequality can be rearranged as  $d_{\text{vul}} \gtrsim \frac{\alpha - \beta}{\beta} d_{\text{rob}}$ . In typical cases where the scale of the robust features is much larger than that of the non-robust features ( $\alpha \gg \beta$ ), we can informally conclude:

**(Informal restatement of Theorem 3.5)** Assume that the scale of the robust features is much larger than that of the non-robust features ( $\alpha \gg \beta$ ), the perturbation size is on the same scale as the non-robust features ( $\epsilon \lesssim \beta$ ), and there are no non-predictive features ( $d_{\text{irr}} = 0$ ). If  $d_{\text{vul}} \gtrsim \frac{\alpha}{\beta} d_{\text{rob}}$ , then the standardly pretrained single-layer linear transformer is vulnerable to adversarial attacks.

*Non-predictive features accelerate vulnerability.* Redundant dimensions  $d_{\text{irr}}$  do not contribute to the first term (i.e., accuracy) but increase the second term (i.e., vulnerability). Therefore, they degrade robustness without improving predictive performance. In addition,  $d_{\text{irr}}$  amplifies the adversarial effect at a rate of  $d_{\text{irr}}\epsilon$ , which is comparable to the effect of the useful dimensions,  $d_{\text{rob}}\epsilon$  and  $d_{\text{vul}}\epsilon$ .

**Adversarial Pretraining Leads to Universal Robustness.** We now establish universal robustness of the adversarially pretrained model.

**Theorem 3.6** (Adversarial pretraining case). Suppose that  $q_{\text{rob}}$  and  $q_{\text{vul}}$  defined in Assumption 3.2 are sufficiently small. There exist constants  $C_1, C_2 > 0$  such that

$$\begin{aligned} & \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1} \sim \mathcal{D}^{\text{te}}} \left[ \min_{\|\Delta\|_\infty \leq \epsilon} y_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}})]_{d+1, N+1} \right] \\ & \geq \underbrace{C_1 (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) (d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2)}_{\text{Prediction for original data}} \\ & \quad - \underbrace{C_2 \left\{ (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \left( d_{\text{rob}}\alpha + d_{\text{vul}}\beta + \frac{d_{\text{irr}}\gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right\}}_{\text{Adversarial effect}} \epsilon. \quad (9) \end{aligned}$$

The proof and a more general statement can be found in Appendix F and Theorem F.1, respectively. For notational simplicity, we assume small  $q_{\text{rob}}$  and  $q_{\text{vul}}$ . However, we do not require infinitesimal  $q_{\text{rob}}$  and  $q_{\text{vul}}$ . See Theorem F.1 and Appendix C. In contrast to Theorem 3.5, this theorem provides a lower bound. A positive right-hand side implies correct classification under adversarial perturbations.

*Adversarially trained models prioritize robust features.* Assume  $d_{\text{irr}} = 0$ . Up to constant factors, the lower bound reduces to  $(d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \{d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2 - (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)\epsilon\}$ . The important factor is  $d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2 - (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)\epsilon$ , which determines the sign. As shown in Theorem 3.5, the standard model extracts the robust and non-robust features at scales  $d_{\text{rob}}\alpha$  and  $d_{\text{vul}}\beta$ , respectively. In contrast, the adversarially trained model extracts them at quadratic scales  $d_{\text{rob}}\alpha^2$  and  $d_{\text{vul}}\beta^2$ . Since the robust features typically have larger magnitude ( $\alpha^2 \gg \beta^2$ ), the adversarially trained model places greater emphasis on the robust features and mitigates the influence of the non-robust features, compared to its standard counterpart.

*Adversarially trained models are universally robust.* As shown above, up to constant factors, the prediction remains correct as long as  $d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2 - (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)\epsilon \geq 0$ . This condition fails when  $\epsilon \gtrsim \frac{d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2}{d_{\text{rob}}\alpha + d_{\text{vul}}\beta}$ . When the perturbation size  $\epsilon$  is on the same scale as the non-robust features ( $\epsilon \lesssim \beta$ ), the inequality can be rearranged as  $d_{\text{vul}} \gtrsim \frac{d_{\text{rob}}\alpha(\alpha - \beta)}{\beta^2}$ . In typical cases where  $\alpha \gg \beta$ , we can informally conclude:

**(Informal restatement of Theorem 3.6)** Assume that the scale of the robust features is much larger than that of the non-robust features ( $\alpha \gg \beta$ ), the perturbation size is on the same scale as the non-robust features ( $\epsilon \lesssim \beta$ ), and there are no non-predictive features ( $d_{\text{irr}} = 0$ ). If  $d_{\text{vul}} \gtrsim \left(\frac{\alpha}{\beta}\right)^2 d_{\text{rob}}$ , then the adversarially pretrained single-layer linear transformer is robust to adversarial attacks.

This threshold substantially improves on the standard model’s robustness condition. For example, when  $\alpha = 160/255$  and  $\beta = 8/255$ , the standard model is potentially robust up to  $d_{\text{vul}} \lesssim 20d_{\text{rob}}$ , whereas the adversarially pretrained model remains robust up to  $d_{\text{vul}} \lesssim 400d_{\text{rob}}$ . This result also suggests that the adversarially pretrained model is potentially vulnerable when the non-robust dimensions significantly outnumber the robust ones, consistent with the failure case in Section 3.3.

*Adversarially trained models are more robust to attacks that exploit non-predictive features.* **Theorem 3.6** shows that even when the adversary exploits redundant dimensions, their effect is significantly attenuated. For simplicity, assume  $N \rightarrow \infty$ . The adversarial effect from the irrelevant features scales as  $d_{\text{irr}}\gamma^2\epsilon$ , which is linear in  $d_{\text{irr}}$ . In contrast, the clean term scales as  $d_{\text{rob}}^2\alpha^3$  and  $d_{\text{vul}}^2\beta^3$ , i.e., quadratically in the number of the informative features. Thus, as long as the informative features dominate in magnitude and number, the influence of the non-predictive features on the model’s robustness remains limited.

### 3.5 OPEN CHALLENGES

In this section, we show that two open challenges in robust classification (Schmidt et al., 2018; Tsipras et al., 2019) persist in our setting.

**Accuracy–Robustness Trade-Off.** Inspired by Tsipras et al. (2019), we consider a situation where robust features correlate with their labels with some probability, whereas non-robust features always correlate.

**Theorem 3.7** (Accuracy–robustness trade-off). Assume  $d_{\text{rob}} = 1$ ,  $d_{\text{vul}} = d - 1$ , and  $d_{\text{irr}} = 0$ . In addition to **Assumption 3.2**, for  $(\mathbf{x}, y) \sim \mathcal{D}^{\text{te}}$ , suppose that  $yx_i$  takes  $\alpha$  with probability  $p > 0.5$  and  $-\alpha$  with probability  $1 - p$  for  $i \in \mathcal{S}_{\text{rob}}$ . Moreover,  $yx_i$  takes  $\beta$  with probability one for  $i \in \mathcal{S}_{\text{vul}}$ . Define  $\tilde{f}(\mathbf{P}, \mathbf{Q}) := \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}^{\text{te}}} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}, \mathbf{Q})]_{d+1, N+1}]$ . Then, there exist strictly positive functions  $g_1(d, \alpha, \beta)$  and  $g_2(d, \alpha, \beta)$  such that

$$\tilde{f}(\mathbf{P}^{\text{std}}, \mathbf{Q}^{\text{std}}) = \begin{cases} g_1(d, \alpha, \beta)(\alpha + (d - 1)\beta) & (\text{w.p. } p) \\ g_1(d, \alpha, \beta)(-\alpha + (d - 1)\beta) & (\text{w.p. } 1 - p) \end{cases}, \quad (10)$$

$$\tilde{f}(\mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}}) \leq g_2(d, \alpha, \beta)\{-(2p - 1)\alpha^2 + (d - 1)\beta^2\} \quad (\text{w.p. } 1 - p). \quad (11)$$

The proof can be found in **Appendix G**. Unlike **Theorems 3.5** and **3.6**, this theorem considers the expectation over  $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ , instead of  $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}$ . The query  $(\mathbf{x}_{N+1}, y_{N+1})$  is stochastic. If  $d \gtrsim \frac{\alpha}{\beta}$ , the standard model consistently produces correct predictions. However, if  $d \lesssim (2p - 1)(\frac{\alpha}{\beta})^2$ , the adversarially trained model produces incorrect predictions with probability  $1 - p$ . This trade-off arises because the adversarially trained model discards the non-robust but predictive features.

**Need for Larger In-Context Sample Sizes.** We informally summarize **Theorem H.1** as follows (omitting constant factors for clarity):

**(Informal summary of Theorem H.1)** Assume the same conditions as in **Theorem 3.7**. Consider  $\mathbb{E}_{\mathbf{x}_{N+1}, y_{N+1}} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}, \mathbf{Q})]_{d+1, N+1}]$ . Assume  $d \lesssim \frac{\alpha}{\beta}$ ,  $p \rightarrow 0.5$ , and a small  $N$  regime. With probability at least  $1 - \exp(-N)$ , the standard model makes correct predictions. With probability at most  $1 - \frac{1}{\sqrt{N}}$ , the adversarially trained model makes correct predictions.

This result indicates that in low-sample regimes, the adversarially pretrained model requires substantially more in-context demonstrations to achieve clean accuracy comparable to that of the standard model. This stems from the model’s reliance on the robust features, which are statistically underrepresented in small-sample regimes.

## 4 EXPERIMENTAL RESULTS

Additional results and detailed experimental settings are provided in **Appendix D**.

**Verification of Theorem 3.4.** We trained single-layer linear transformers (2) using stochastic gradient descent over  $[0, 1]^d$  with the in-context loss (7). The training distribution was configured with  $d = 20$  and  $\lambda = 0.1$ . We used  $\epsilon = 0$ ,  $\frac{1+(d-1)(\lambda/2)}{d} = 0.098$ , and  $\frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3} = 0.95$  for the standard, adversarial, and strongly adversarial regimes, respectively. The heatmaps of the learned parameters are shown in **Fig. 1**. These results align with the theoretical predictions of **Theorem 3.4**.

**Verification of Theorems 3.5 to 3.7.** We evaluated the standardly and adversarially pretrained single-layer linear transformers with the theoretically predicted parameters (i.e., the parameters in

Table 1: Accuracy (%) of standardly and adversarially pretrained single-layer linear transformers. Left values represent clean accuracy. Right values represent robust accuracy. For  $\mathcal{D}^{\text{tr}}$  (cf. [Assumption 3.1](#)), we used  $d = 100$  and  $\lambda = 0.1$ . For  $\mathcal{D}^{\text{te}}$  (cf. [Assumption 3.2](#)), we constructed the test distribution from multivariate normal distributions with  $d_{\text{rob}} = 10$ ,  $d_{\text{vul}} = 90$ ,  $d_{\text{irr}} = 0$ ,  $\alpha = 1.0$ , and  $\beta = 0.1$ . For the real-world datasets, the values were averaged across all 45 binary classification pairs from the 10 classes. The perturbation budgets were set as follows:  $\epsilon = 0.15$  for  $\mathcal{D}^{\text{tr}}$ , 0.2 for  $\mathcal{D}^{\text{te}}$ , 0.1 for MNIST and CIFAR-10, and 0.15 for Fashion-MNIST. See [Appendix D](#) for details.

	$\mathcal{D}^{\text{tr}}$	$\mathcal{D}^{\text{te}}$	MNIST	FMNIST	CIFAR10
Standardly pretrained model	<b>100</b> / 0	<b>100</b> / 0	<b>94</b> / 4	<b>91</b> / 20	<b>68</b> / 21
Adversarially pretrained model	<b>100</b> / <b>100</b>	99 / <b>95</b>	93 / <b>72</b>	89 / <b>62</b>	64 / <b>34</b>

the standard regime and adversarial regime in [Theorem 3.4](#) on  $\mathcal{D}^{\text{tr}}$ ,  $\mathcal{D}^{\text{te}}$ , MNIST ([Deng, 2012](#)), Fashion-MNIST ([Xiao et al., 2017](#)), and CIFAR-10 ([Krizhevsky, 2009](#)). These results are provided in [Tab. 1](#). The results suggest that the standard models achieve high clean accuracy but suffer severe degradation under adversarial attacks, consistent with [Theorem 3.5](#). In contrast, the adversarially pretrained models maintain high robustness, supporting [Theorem 3.6](#), while their clean accuracy is lower, aligning with the accuracy–robustness trade-off described in [Theorem 3.7](#).

## 5 CONCLUSION AND LIMITATIONS

We theoretically demonstrated that single-layer linear transformers, after adversarial pretraining across classification tasks, can robustly adapt to previously unseen classification tasks through in-context learning, without any additional training. These results pave the way for universally robust foundation models. We also showed that these transformers can adaptively focus on robust features, exhibit the accuracy–robustness trade-off, and require a larger number of in-context demonstrations.

Our limitations include assumptions about data distributions and architectures. While we assume that the data distributions consist of clearly separated robust and non-robust features, real-world datasets typically exhibit a more gradual transition (cf. [Section 3.1](#), especially Example 3). Single-layer linear transformers lack the practical characteristics of multi-layer models and softmax attention. Although these theoretical assumptions are standard and comparable in strength to those in prior work (cf. studies on in-context learning in [Appendix A](#)), they limit the applicability of our results.

Universally robust foundation models are conceptually expected to adapt to any task and any form of perturbations. However, our theoretical results assume classification tasks and  $\ell_\infty$  perturbations. Extending these results to other tasks and perturbation models is left for future work.

The cost of adversarial pretraining is also a limitation of universally robust foundation models. We expect that such efforts will be undertaken by large organizations, which could offset development costs through API fees. In addition, acceleration techniques for adversarial training, which have been extensively studied in the literature, can reduce this cost to a level comparable to standard training. Our theoretical analysis is an important first step toward fostering the practical development of universally robust foundation models. See also the last paragraph in [Section 1](#).

### REPRODUCIBILITY STATEMENT

All experimental procedures are described in [Section 4](#) and [Appendix D](#). The source code to reproduce our experimental results can be found in <https://github.com/s-kumano/universally-robust-in-context-learner>. Proofs of the theorems are provided in [Appendices E](#) to [H](#).

### THE USE OF LARGE LANGUAGE MODELS (LLMs)

We used LLMs to improve our writing. No essential contributions were made by the LLMs.

#### ACKNOWLEDGMENTS AND DISCLOSURE OF FUNDING

S. Kumano was supported by JSPS KAKENHI Grant Number JP23KJ0789 and by JST, ACT-X Grant Number JPMJAX23C7, JAPAN. H. Kera was supported by JST PRESTO Grant Number JPMJPR24K, JST BOOST Program Grant Number JPMJBY24C6, and JSPS Program for Forming Japan’s Peak Research Universities (J-PEAKS) Grant Number JPJS00420230002.

#### REFERENCES

- Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *NeurIPS*, volume 36, pp. 45614–45650, 2023.
- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *ICLR*, 2023.
- Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Deforges. Reveal of vision transformers robustness against adversarial attacks. *arXiv:2106.03734*, 2021.
- Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *NeurIPS*, volume 33, pp. 16048–16059, 2020.
- Usman Anwar, Johannes Von Oswald, Louis Kirsch, David Krueger, and Spencer Frei. Adversarial robustness of in-context learning in transformers for linear regression. *arXiv:2411.05189*, 2024.
- Robert B Ash. *Information Theory*. Courier Corporation, 1990.
- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pp. 274–283, 2018.
- Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in- and out-distribution improves explainability. In *ECCV*, pp. 228–245, 2020.
- Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *NeurIPS*, volume 36, pp. 57125–57211, 2023.
- Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? In *NeurIPS*, volume 34, pp. 26831–26843, 2021.
- Philipp Benz, Soomin Ham, Chaoning Zhang, Adil Karjauv, and In So Kweon. Adversarial robustness comparison of vision transformer and MLP-mixer to CNNs. In *BMVC*, 2021.
- Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *ICCV*, pp. 10231–10241, 2021.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, volume 33, pp. 1877–1901, 2020.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security*, pp. 2633–2650, 2021.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *ICLR*, 2022.
- Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *ICML*, pp. 1383–1391, 2020.
- Xiang Cheng, Yuxin Chen, and Suvrit Sra. Transformers implement functional gradient descent to learn non-linear functions in context. In *ICML*, 2024.

- Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, pp. 2206–2216, 2020.
- Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ACL*, 2023.
- Edoardo DeBenedetti, Vikash Sehwal, and Prateek Mittal. A light recipe to train robust vision transformers. In *SaTML*, pp. 225–253, 2023.
- Li Deng. The MNIST database of handwritten digit images for machine learning research. *Signal Processing Magazine*, 29(6):141–142, 2012.
- Edgar Dobriban, Hamed Hassani, David Hong, and Alexander Robey. Provable tradeoffs in adversarially robust classification. *IEEE Transactions on Information Theory*, 2023.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv:1906.00945*, 2019.
- Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. In *ICML*, 2019.
- Hao Fan, Zhaoyang Ma, Yong Li, Rui Tian, Yunli Chen, and Chenlong Gao. Mixprompt: Enhancing generalizability and adversarial robustness for vision-language models via prompt fusion. In *ICIC*, pp. 328–339, 2024.
- Spencer Frei and Gal Vardi. Trained transformer classifiers generalize and exhibit benign overfitting in-context. In *ICLR*, 2025.
- Deqing Fu, Tian-qi Chen, Robin Jia, and Vatsal Sharan. Transformers learn to achieve second-order convergence rates for in-context linear regression. In *NeurIPS*, volume 37, pp. 98675–98716, 2024.
- Shaopeng Fu, Liang Ding, and Di Wang. ”short-length” adversarial training helps llms defend ”long-length” jailbreak attacks: Theoretical and empirical evidence. *arXiv:2502.04204*, 2025.
- Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *NeurIPS*, volume 35, pp. 30583–30598, 2022.
- Siddhant Garg and Goutham Ramakrishnan. BAE: BERT-based adversarial examples for text classification. In *EMNLP*, pp. 6174–6181, 2020.
- Khashayar Gatmiry, Nikunj Saunshi, Sashank J Reddi, Stefanie Jegelka, and Sanjiv Kumar. Can looped transformers learn to implement multi-step gradient descent for in-context learning? In *ICML*, 2024.
- Angeliki Giannou, Liu Yang, Tianhao Wang, Dimitris Papailiopoulos, and Jason D Lee. How well can transformers emulate in-context newton’s method? *arXiv:2403.03183*, 2024.
- Micah Goldblum, Liam Fowl, and Tom Goldstein. Adversarially robust few-shot learning: A meta-learning approach. In *NeurIPS*, volume 33, pp. 17886–17895, 2020.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- Yufan Hou, Lixin Zou, and Weidong Liu. Task-based focal loss for adversarially robust meta-learning. In *ICPR*, pp. 2824–2829, 2021.
- Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, volume 32, pp. 125–136, 2019.

- Ahmadreza Jeddi, Mohammad Javad Shafiee, and Alexander Wong. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *arXiv:2012.13628*, 2020.
- Xiaojun Jia, Sensen Gao, Simeng Qin, Ke Ma, Xinfeng Li, Yihao Huang, Wei Dong, Yang Liu, and Xiaochun Cao. Evolution-based region adversarial prompt learning for robustness enhancement in vision-language models. *arXiv:2503.12874*, 2025.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*, volume 34, pp. 8018–8025, 2020.
- Simran Kaur, Jeremy Cohen, and Zachary C Lipton. Are perceptually-aligned gradients a general property of robust classifiers? In *NeurIPS WS*, 2019.
- Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *AAAI*, volume 35, pp. 8119–8127, 2021.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *NeurIPS*, volume 36, pp. 43057–43083, 2023.
- Lin Li, Haoyan Guan, Jianing Qiu, and Michael Spratling. One prompt word is enough to boost adversarial robustness for pre-trained vision-language models. In *CVPR*, pp. 24408–24419, 2024.
- Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *EMNLP*, pp. 6193–6202, 2020.
- Tianle Li, Chenyang Zhang, Xingwu Chen, Yuan Cao, and Difan Zou. On the robustness of transformers against context hijacking for linear classification. *arXiv:2502.15609*, 2025.
- Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *ICLR*, 2024.
- Chang Liu, Yinpeng Dong, Wenzhao Xiang, Xiao Yang, Hang Su, Jun Zhu, Yuefeng Chen, Yuan He, Hui Xue, and Shibao Zheng. A comprehensive study on robustness of image classification models: Benchmarking and rethinking. *IJCV*, 133(2):567–589, 2025.
- Fan Liu, Shuyu Zhao, Xuelong Dai, and Bin Xiao. Long-term cross adversarial training: A robust meta-learning method for few-shot classification tasks. In *ICML WS*, 2021.
- Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR*, 2024.
- Lin Luo, Xin Wang, Bojia Zi, Shihao Zhao, and Xingjun Ma. Adversarial prompt distillation for vision-language models. *arXiv:2411.15244*, 2024.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *ICLR*, 2024.
- Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *ICCV*, pp. 7838–7847, 2021.
- Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *ICLR*, 2023.
- Mohammad Mehrabi, Adel Javanmard, Ryan A Rossi, Anup Rao, and Tung Mai. Fundamental tradeoffs in distributionally adversarial training. In *ICML*, pp. 7544–7554, 2021.

- Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *NeurIPS*, volume 34, pp. 23296–23308, 2021.
- Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv:2311.17035*, 2023.
- Geon Yeong Park and Sang Wan Lee. Reliably fast adversarial training via latent adversarial perturbation. In *ICCV*, pp. 7758–7767, 2021.
- Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *AAAI*, volume 36, pp. 2071–2081, 2022.
- Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. In *NeurIPS WS*, 2022.
- Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. In *ICML WS*, 2019.
- Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, 2020.
- Daniel L Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4): 517, 1994.
- Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, volume 32, 2019.
- Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, volume 31, 2018.
- Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.
- Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *TMLR*, 2022.
- Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. ”Do anything now”: Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *ACM CCS*, pp. 1671–1685, 2024.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *ICML*, pp. 31210–31227, 2023.
- Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? In *ICML*, 2024.
- Suraj Srinivas, Sebastian Bordt, and Himabindu Lakkaraju. Which models have perceptually-aligned gradients? an explanation via off-manifold robustness. In *NeurIPS*, volume 36, 2023.
- Anuj Srivastava, Ann B Lee, Eero P Simoncelli, and S-C Zhu. On advances in statistical modeling of natural images. *Journal of mathematical imaging and vision*, 18(1):17–33, 2003.
- Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?—a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, pp. 631–648, 2018.
- Satoshi Suzuki, Shin’ya Yamaguchi, Shoichiro Takeda, Sekitoshi Kanai, Naoki Makishima, Atsushi Ando, and Ryo Masumura. Adversarial finetuning with latent representation constraint to mitigate accuracy-robustness tradeoff. In *ICCV*, pp. 4367–4378, 2023.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

- Shiyu Tang, Ruihao Gong, Yan Wang, Aishan Liu, Jiakai Wang, Xinyun Chen, Fengwei Yu, Xianglong Liu, Dawn Song, Alan Yuille, et al. RobustART: Benchmarking robustness on architecture design and training techniques. *arXiv:2109.05211*, 2021.
- Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, volume 33, pp. 1633–1645, 2020.
- Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *ICML*, pp. 35151–35174, 2023.
- Johannes Von Oswald, Maximilian Schlegel, Alexander Meulemans, Seijin Kobayashi, Eyvind Niklasson, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Max Vladymyrov, et al. Uncovering mesa-optimization algorithms in transformers. In *ICLR WS*, 2024.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP-IJCNLP*, 2019.
- Ren Wang, Kaidi Xu, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Chuang Gan, and Meng Wang. On fast adversarial robustness adaptation in model-agnostic meta-learning. In *ICLR*, 2021.
- Sibo Wang, Jie Zhang, Zheng Yuan, and Shiguang Shan. Pre-trained model guided fine-tuning for zero-shot adversarial robustness. In *CVPR*, pp. 24502–24511, 2024a.
- Xin Wang, Kai Chen, Xingjun Ma, Zhineng Chen, Jingjing Chen, and Yu-Gang Jiang. Advqdet: Detecting query-based adversarial attacks with adversarial contrastive prompt tuning. In *ACM MM*, pp. 6212–6221, 2024b.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *NeurIPS*, volume 36, pp. 80079–80110, 2023a.
- Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv:2303.03846*, 2023b.
- Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. In *NeurIPS*, volume 36, pp. 36637–36651, 2023.
- Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.
- Boxi Wu, Jindong Gu, Zhifeng Li, Deng Cai, Xiaofei He, and Wei Liu. Towards efficient adversarial training on vision transformers. In *ECCV*, pp. 307–325, 2022.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.
- Fan Yang, Mingxuan Xia, Sangzhou Xia, Chicheng Ma, and Hui Hui. Revisiting the robust generalization of adversarial prompt tuning. *arXiv:2405.11154*, 2024.
- Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *NeurIPS*, volume 33, pp. 8588–8601, 2020.
- Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Adversarial meta-learning. *arXiv:1806.03316*, 2018.
- Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *ACL*, pp. 6066–6080, 2020.
- Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *NeurIPS*, 2019a.

- Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pp. 7472–7482, 2019b.
- Jiaming Zhang, Xingjun Ma, Xin Wang, Lingyu Qiu, Jiaqi Wang, Yu-Gang Jiang, and Jitao Sang. Adversarial prompt tuning for vision-language models. In *ECCV*, pp. 56–72, 2024a.
- Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *JMLR*, 25(49):1–55, 2024b.
- Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *ICML*, pp. 7502–7511, 2019.
- Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv:2305.19420*, 2023.
- Yiwei Zhou, Xiaobo Xia, Zhiwei Lin, Bo Han, and Tongliang Liu. Few-shot adversarial prompt learning on vision-language models. In *NeurIPS*, volume 37, pp. 3122–3156, 2024.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv:2307.15043*, 2023.

<b>A Additional Related Work</b>	<b>17</b>
<b>B Clarification on Single-Task Pretraining and Task-Specific Adversarial Training</b>	<b>19</b>
<b>C Additional Theoretical Support and Insights</b>	<b>19</b>
C.1 Linear Combination of Orthonormal Bases can be Transformed into Our Test Distribution. . . . .	19
C.2 Sufficient Number of Datasets to Provide Universal Robustness . . . . .	20
C.3 Effects of $q_{\text{rob}}$ and $q_{\text{vul}}$ . . . . .	20
C.4 Disadvantage of Standard Finetuning: Parameter Selection Perspective . . . . .	20
C.5 Naive Adversarial Context may not Improve Robustness . . . . .	21
<b>D Additional Experimental Results</b>	<b>21</b>
D.1 Support for Assumption 3.2. . . . .	21
D.2 Verification of Theorem 3.4. . . . .	21
D.3 Verification of Theorems 3.5 to 3.7 and H.1 . . . . .	22
<b>E Proof of Lemma 3.3 and Theorem 3.4 (Pretraining)</b>	<b>22</b>
<b>F Proof of Theorems 3.5 and 3.6 (Robustness)</b>	<b>35</b>
<b>G Proof of Theorem 3.7 (Trade-Off)</b>	<b>42</b>
<b>H Proof of Theorem H.1 (Need for Larger Sample Size)</b>	<b>43</b>

## A ADDITIONAL RELATED WORK

**In-Context Learning.** In-context learning has emerged as a remarkable property of large language models, enabling them to adapt to a new task from a few input–output demonstrations without any parameter updates (Brown et al., 2020). Recent work has shown that in-context learning can implement various algorithms (Bai et al., 2023; Garg et al., 2022). One research direction has linked in-context learning with preconditioned gradient descent through empirical (Akyürek et al., 2023; Dai et al., 2023; Garg et al., 2022; Von Oswald et al., 2023; 2024) and theoretical analyses (Ahn et al., 2023; Bai et al., 2023; Cheng et al., 2024; Gatmiry et al., 2024; Mahankali et al., 2024; Zhang et al., 2024b). Additional results have indicated that in-context learning can implement ridge regression (Akyürek et al., 2023; Bai et al., 2023), second-order optimization (Fu et al., 2024; Giannou et al., 2024), reinforcement learning (Lee et al., 2023; Lin et al., 2024), and Bayesian model averaging (Zhang et al., 2023). In terms of robustness, some studies have shown that in-context learning can act as a nearly optimal predictor under noisy linear data (Bai et al., 2023) and noisy labels (Frei & Vardi, 2025). Moreover, it has been demonstrated that in-context learning is robust to shifts in the query distribution (Wies et al., 2023; Zhang et al., 2024b), but not necessarily to shifts in the context (Shi et al., 2023; 2024; Wei et al., 2023b; Zhang et al., 2024b). In this study, we focus on the adversarial robustness of in-context learning, rather than the underlying algorithms or its robustness to random noise and distribution shifts. Specifically, we examine whether a single adversarially pretrained transformer can robustly adapt to a broad range of tasks through in-context learning.

**Norm- and Token-Bounded Adversarial Examples.** Adversarial examples were originally introduced as subtle perturbations to natural data, designed to induce misclassifications in models (Croce & Hein, 2020; Goodfellow et al., 2015; Madry et al., 2018; Szegedy et al., 2014). These perturba-

tions are typically constrained by a norm-based distance from the original inputs. The robustness of transformers to such norm-bounded adversarial examples has been studied primarily in vision transformers (Dosovitskiy et al., 2021). Several studies have shown that standard vision transformers are as vulnerable to these attacks as conventional vision models (Bai et al., 2021; Mahmood et al., 2021), though some have reported marginal differences (Aldahdooh et al., 2021; Benz et al., 2021; Bhojanapalli et al., 2021; Naseer et al., 2021; Paul & Chen, 2022; Shao et al., 2022; Tang et al., 2021). In contrast, adversarial attacks on language models are often neither norm-constrained nor imperceptible to humans. They involve substantial token modifications (Garg & Ramakrishnan, 2020; Jin et al., 2020; Li et al., 2020; Zang et al., 2020), the insertion of adversarial tokens (Liu et al., 2024; Shen et al., 2024; Wallace et al., 2019; Wei et al., 2023a; Zou et al., 2023), and the construction of entirely new adversarial prompts (Carlini et al., 2021; 2022; Nasr et al., 2023; Perez & Ribeiro, 2022; Wei et al., 2023a). These attacks aim not only to induce misclassification (Garg & Ramakrishnan, 2020; Jin et al., 2020; Li et al., 2020; Wallace et al., 2019; Zang et al., 2020), but also to provoke objectionable outputs (Liu et al., 2024; Perez & Ribeiro, 2022; Shen et al., 2024; Wei et al., 2023a; Zou et al., 2023) or to extract private information from training data (Carlini et al., 2021; 2022; Nasr et al., 2023). They are generally bounded by token-level metrics (e.g., the number of modified tokens). In this study, we focus exclusively on norm-bounded adversarial examples. Token-bounded ones are out of scope.

**Adversarial Training.** Adversarial training, which augments training data with adversarial examples, is one of the most effective adversarial defenses (Goodfellow et al., 2015; Madry et al., 2018). Although originally developed for conventional neural architectures, adversarial training has also proven effective for transformers (Debenedetti et al., 2023; Liu et al., 2025; Shao et al., 2022; Tang et al., 2021; Wu et al., 2022). A major limitation of adversarial training is its high computational cost. To address this, several methods have focused on more efficient generation of adversarial examples (Andriushchenko & Flammarion, 2020; Kim et al., 2021; Park & Lee, 2021; Shafahi et al., 2019; Wong et al., 2020; Zhang et al., 2019a) and adversarial finetuning of standard pretrained models (Jeddi et al., 2020; Mao et al., 2023; Suzuki et al., 2023; Wang et al., 2024a). More recently, researchers have introduced adversarial prompt tuning, which trains visual (Mao et al., 2023; Wang et al., 2024b), textual (Fan et al., 2024; Li et al., 2024; Zhang et al., 2024a), or bimodal prompts (Jia et al., 2025; Luo et al., 2024; Yang et al., 2024; Zhou et al., 2024) in an adversarial manner. However, these methods require retraining for each task. In this study, we explore the potential of adversarially pretrained transformers for robust task adaptation via in-context learning, thereby eliminating the task-specific retraining and associated computational overhead.

**Adversarial Meta-Learning.** Adversarial meta-learning seeks to develop a universally robust meta-learner that can swiftly and reliably adapt to new tasks under adversarial conditions. Existing approaches adversarially train a neural network on multiple tasks, and then finetune it on a target task using clean (Goldblum et al., 2020; Hou et al., 2021; Liu et al., 2021; Wang et al., 2021; Yin et al., 2018) or adversarial samples (Yin et al., 2018). In this study, we similarly aim to train such a meta-learner. However, rather than relying on neural networks and finetuning, we employ a transformer as the meta-learner and leverage its in-context learning ability for task adaptation.

**Related but Distinct Work.** We here review theoretical work on the adversarial robustness of in-context learning. Assuming token-bounded adversarial examples, prior studies have shown that even a single token modification in the context can significantly alter the output of a standardly trained model on a clean query (Anwar et al., 2024), and deeper layers can mitigate this (Li et al., 2025). Assuming norm- and token-bounded examples, Fu et al. have shown that adversarial training with short adversarial contexts can provide robustness against longer ones (Fu et al., 2025). They considered a clean query and adversarial tokens appended to the original context. In this study, we explore how adversarially trained models handle norm-bounded perturbations to a query in a clean context. As a result, we reveal their universal robustness that can be generalized to a new task from a few demonstrations.

## B CLARIFICATION ON SINGLE-TASK PRETRAINING AND TASK-SPECIFIC ADVERSARIAL TRAINING

**Single-Task Adversarial Training and In-Context Learning.** We should clarify that a model trained on a single task, unlike one pretrained on multiple tasks (i.e., our main setting), lacks in-context learning capability. Namely, an approach that combines adversarial training (not adversarial pretraining) and in-context learning is not feasible. Specifically, the parameters of a model trained on a single task differ significantly from those on multiple tasks (shown in [Theorem 3.4](#)). Such models cannot provide correct answers for new tasks via in-context learning even in standard settings (without adversaries). The interesting property of transformers is that when trained on multiple tasks rather than a single one—as with large language models—they develop distinctly different parameters that enable in-context learning capability.

**Performance Comparison with Task-Specific Adversarial Training.** As the no-free-lunch theorem indicates, task-specific approaches achieve higher performance than the approach that combines adversarial pretraining and in-context learning. Specifically, the following holds in terms of robust accuracy: (1) task-specific adversarially trained models  $\geq$  (2) adversarially pretrained models with in-context learning  $\gg$  (3) task-specific standardly trained models  $\approx$  (4) standardly pretrained models with in-context learning  $\approx 0\%$ . More precisely, Models (2) have the limitation of robustness as predicted in [Theorems 3.4](#) and [3.6](#). However, if training and test distributions match, Models (1) do not. However, we emphasize that Models (1) require users to perform adversarial training for each individual task and cannot generalize to test distributions other than the one it was trained on. This makes it unsuitable for our research focus on universally robust foundation models that can generalize across a wide range of tasks without task-specific adversarial training.

## C ADDITIONAL THEORETICAL SUPPORT AND INSIGHTS

### C.1 LINEAR COMBINATION OF ORTHONORMAL BASES CAN BE TRANSFORMED INTO OUR TEST DISTRIBUTION.

Our test data distribution, [Assumption 3.2](#), can implicitly represent data distributions comprising robust and non-robust directions forming an orthonormal basis. Consider  $d$  orthonormal bases,  $\{e_i\}_{i=1}^d$ . We set  $d_{\text{irr}} = 0$ , namely  $d = d_{\text{rob}} + d_{\text{vul}}$ . Each data point is represented as  $\mathbf{x} = c_1 e_1 + c_2 e_2 + \dots + c_d e_d$ , where coefficients  $c_i$  are sampled probabilistically. These coefficients satisfy  $\mathbb{E}[y c_i] = C_i \alpha$  for  $i \in \mathcal{S}_{\text{rob}}$  and  $\beta$  for  $i \in \mathcal{S}_{\text{vul}}$ . In addition,  $|\mathbb{E}[(y c_i - \mathbb{E}[y c_i])^n]| \leq C_{i,n} \alpha^n$  for  $i \in \mathcal{S}_{\text{rob}}$  and  $C_{i,n} \beta^n$  for  $i \in \mathcal{S}_{\text{vul}}$ . Given a dataset of  $N$  i.i.d. samples  $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ , if  $c_{n,i}$  is independent of  $c_{n,j}$  for  $i \neq j$  conditional on  $y$ , and  $N$  is sufficiently large, then the covariance of  $y\mathbf{x}$  can be approximated as:

$$\begin{aligned} & \frac{1}{N} \sum_{n=1}^N \left( y_n \mathbf{x}_n - \sum_{k=1}^N y_k \mathbf{x}_k \right) \left( y_n \mathbf{x}_n - \sum_{k=1}^N y_k \mathbf{x}_k \right)^\top \\ & \approx \mathbb{E}[(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])^\top] \end{aligned} \quad (\text{A12})$$

$$= \mathbb{E} \left[ \left( \sum_{i=1}^d (y_i c_i - \mathbb{E}[y c_i]) e_i \right) \left( \sum_{i=1}^d (y_i c_i - \mathbb{E}[y c_i]) e_i \right)^\top \right] \quad (\text{A13})$$

$$= \sum_{i,j=1}^d \mathbb{E}[(y c_i - \mathbb{E}[y c_i])(y c_j - \mathbb{E}[y c_j])] e_i e_j^\top \quad (\text{A14})$$

$$= \sum_{i \in \mathcal{S}_{\text{rob}}} C_{i,2} \alpha^2 e_i e_i^\top + \sum_{i \in \mathcal{S}_{\text{vul}}} C_{i,2} \beta^2 e_i e_i^\top. \quad (\text{A15})$$

This implies that through principal component analysis for  $y_n \mathbf{x}_n$ , we can obtain  $d$  orthonormal bases,  $\{e_i\}_{i=1}^d$ . By projecting a sample  $\mathbf{x}_n$  onto these bases, we obtain a transformed sample  $\mathbf{x}'_n := \{c_{n,1}, c_{n,2}, \dots, c_{n,d}\}$ . This demonstrates that when data is sampled from a distribution comprising robust and non-robust directions forming an orthonormal basis, if the coefficients are

mutually independent and the sample size is sufficiently large, we can preprocess the data to satisfy [Assumption 3.2](#). Importantly, this preprocessing relies solely on statistics derivable from training samples.

## C.2 SUFFICIENT NUMBER OF DATASETS TO PROVIDE UNIVERSAL ROBUSTNESS

What determines the sufficient number of datasets needed to provide universal robustness to transformers? We conjecture that this may be determined by the number of robust bases. In this paper, we trained transformers using  $d$  datasets. This stems from training with datasets where only one dimension is robust (in other words, datasets with a single robust basis), the number of dimensions  $d$ , and the assumption that all dimensions might contain robust features. If we assume that robust features never appear in the latter  $d'$  dimensions, following the procedure in [Appendix E](#), we can train robust transformers using only  $d - d'$  datasets that describe the first  $d - d'$  robust features. From this observation, we conjecture that the sufficient number of datasets required to provide universal robustness to transformers depends on the number of robust bases in the assumed data structure.

## C.3 EFFECTS OF $q_{\text{rob}}$ AND $q_{\text{vul}}$

We here analyze how  $q_{\text{rob}}$  and  $q_{\text{vul}}$  affect the robustness of adversarially trained transformer. As defined in [Assumption 3.2](#), these parameters control the proportion of features whose total covariance with other features is negative. [Theorem F.1](#) suggests that the transformer prediction for unperturbed data can be expressed as

$$C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta)\{(1 - cq_{\text{rob}})d_{\text{rob}}\alpha^2 + (1 - cq_{\text{vul}})d_{\text{vul}}\beta^2\} + C'(d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2), \quad (\text{A16})$$

where

$$c := \frac{(\max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i)(\max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_{i,2})}{\min_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i^3}. \quad (\text{A17})$$

Examining the term  $(1 - cq_{\text{rob}})d_{\text{rob}}\alpha^2 + (1 - cq_{\text{vul}})d_{\text{vul}}\beta^2$ , we observe that larger values of  $q_{\text{rob}}$  and  $q_{\text{vul}}$  generally diminish the magnitude of transformer predictions. This indicates that negative correlations between features degrade the robustness of adversarially trained transformers. Additionally, the coefficient  $c$  is characterized by  $\max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_{i,2}$ , which represents a variance coefficient. This suggests that smaller feature variances enhance the robustness of adversarially trained transformers. For example, if each feature variance  $C_{i,2}$  is sufficiently small, even  $q_{\text{rob}} = 1$  and  $q_{\text{vul}} = 1$  may be tolerated without significantly compromising robustness.

## C.4 DISADVANTAGE OF STANDARD FINETUNING: PARAMETER SELECTION PERSPECTIVE

In this study, we investigate task adaptation through in-context learning. As an alternative lightweight approach, standard finetuning—where all or part of the model parameters are updated—can also be employed. However, a key drawback of standard finetuning is that it requires parameter updates, whereas in-context learning does not. Moreover, finetuning necessitates careful selection of which parameters to update. Our analysis shows that improper parameter selection during finetuning can compromise the robustness initially established by adversarial pretraining. Consider adversarially pretrained parameters,  $\mathbf{P}^{\text{adv}}$  and  $\mathbf{Q}^{\text{adv}}$ , and  $\mathcal{D}_c^{\text{tr}}$  as a downstream data distribution.

First, we examine the scenario where only  $\mathbf{P}$  is updated while keeping  $\mathbf{Q}^{\text{adv}}$  fixed, formulated as:

$$\min_{\mathbf{P} \in [0,1]^{(d+1) \times (d+1)}} \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}, \mathcal{D}_c^{\text{tr}}} [-y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}, \mathbf{Q}^{\text{adv}})]_{d+1, N+1}]. \quad (\text{A18})$$

In this case, as shown in the proof in [Appendix E](#),  $\mathbf{P} = \mathbf{P}^{\text{std}} (= \mathbf{P}^{\text{adv}})$  is the global solution. Consequently, as demonstrated in [Theorem 3.6](#), the model’s robustness is preserved.

Conversely, consider training  $\mathbf{Q}$  while keeping  $\mathbf{P}^{\text{adv}}$  fixed, formulated as:

$$\min_{\mathbf{Q} \in [0,1]^{(d+1) \times (d+1)}} \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}, \mathcal{D}_c^{\text{tr}}} [-y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}^{\text{adv}}, \mathbf{Q})]_{d+1, N+1}]. \quad (\text{A19})$$

In this scenario,  $\mathbf{Q} = \mathbf{Q}^{\text{std}}$  is the global solution. As established in [Theorems 3.5, 3.7](#) and [H.1](#), while this configuration enables the transformer to perform well on unperturbed queries, it fails to maintain robustness against perturbed inputs.

These findings highlight a critical insight: achieving robust task adaptation through standard finetuning requires careful parameter selection; otherwise, the pretrained model’s adversarial robustness may be compromised. This parameter sensitivity represents a disadvantage compared to in-context learning, which preserves robustness without requiring parameter updates.

### C.5 NAIVE ADVERSARIAL CONTEXT MAY NOT IMPROVE ROBUSTNESS

One approach to enhancing the robustness of a standardly trained transformer is to incorporate adversarial examples into the context. In this section, we show that this is not the case in our setting. Consider the following transformer input:

$$\mathbf{Z}' := \begin{bmatrix} \mathbf{x}_1 + \Delta_1 & \mathbf{x}_2 + \Delta_2 & \cdots & \mathbf{x}_N + \Delta_N & \mathbf{x}_{N+1} + \Delta_{N+1} \\ y_1 & y_2 & \cdots & y_N & 0 \end{bmatrix}. \quad (\text{A20})$$

The adversarial perturbations for the context,  $\Delta_1, \dots, \Delta_N$ , are defined as  $\Delta_n := -\epsilon y_n \mathbf{1}_d$ . In this setting, for  $\epsilon \geq \frac{1+(d-1)(\lambda/2)}{d}$ , the standard transformer prediction is given by:

$$\mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1} \sim \mathcal{D}_c^{\text{tr}}} \left[ \min_{\|\Delta_{N+1}\|_\infty \leq \epsilon} y_{N+1} [f(\mathbf{Z}'; \mathbf{P}^{\text{std}}, \mathbf{Q}^{\text{std}})]_{d+1, N+1} \right] \leq 0. \quad (\text{A21})$$

This result suggests that, in our setting, naive adversarial demonstrations do not improve the performance of the standard transformer. Intuitively, because adversarial training generates new adversarial examples at each step of gradient descent, fixed adversarial demonstrations may fail to counter newly generated adversarial perturbations to the query.

## D ADDITIONAL EXPERIMENTAL RESULTS

All experiments were conducted on Ubuntu 20.04.6 LTS, Intel Xeon Gold 6226R CPUs, and NVIDIA RTX 6000 Ada GPUs.

### D.1 SUPPORT FOR ASSUMPTION 3.2.

The statistics of preprocessed MNIST, Fashion-MNIST, and CIFAR-10 are provided in Fig. A2. Preprocessing was conducted as follows: (i) selection of two different classes from the ten available classes and assignment of binary labels to every sample from the training dataset, creating  $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ ; (ii) centering the data via  $\mathbf{x}' \leftarrow \mathbf{x} - \bar{\mathbf{x}}$  with  $\bar{\mathbf{x}} := (1/N) \sum_{n=1}^N \mathbf{x}_n$ ; and (iii) aligning features with the label using  $\mathbf{x}'' \leftarrow \text{sgn}(\sum_{n=1}^N y_n \mathbf{x}_n) \odot \mathbf{x}'$ . These preprocessed datasets exhibit that each dimension has a positive correlation with the label and that few dimensions have negative total covariance. The main distinction from Assumption 3.2 is that their features are not clearly separated as robust or non-robust. Instead, they gradually transition from robust to non-robust characteristics.

### D.2 VERIFICATION OF THEOREM 3.4.

We trained a single-layer transformer (2) with the in-context loss (7). The training distribution was configured with  $d = 20$  and  $\lambda = 0.1$  in Fig. 1 and with  $d = 100$  and  $\lambda = 0.1$  in Fig. A3. For standard, adversarial, and strong adversarial regimes, we used  $\epsilon = 0$ ,  $\frac{1+(d-1)(\lambda/2)}{d} = 0.098$ , and  $\frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3} = 0.95$  in Fig. 1 and  $\epsilon = 0$ ,  $\frac{1+(d-1)(\lambda/2)}{d} = 0.06$ , and  $\frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3} = 0.77$  in Fig. A3. Optimization was conducted using stochastic gradient descent with momentum 0.9. Learning rates were set to 0.1 for all regimes in Fig. 1, and to 1.0 for standard and strong adversarial regimes and 0.2 for the adversarial regime in Fig. A3. Training ran for 100 epochs with a learning rate scheduler that multiplied the rate by 0.1 when the loss did not improve within 10 epochs. In each iteration of stochastic gradient descent, we sampled 1,000 datasets  $\{(\mathbf{x}_n^{(c)}, y_n^{(c)})\}_{n=1}^{N+1}$  with  $N = 1,000$ . The distribution index  $c$  was randomly sampled from  $U([d])$ , meaning that in each iteration, each of the 1,000 datasets may have different  $c$  values. After each parameter update, we projected the parameters to  $[0, 1]^d$ . Adversarial perturbation was calculated as  $\Delta := -\epsilon y_n \text{sgn}(\mathbf{P}_{d+1}, \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{Q}_{\cdot, :d})$ , which represents the optimal attack. The heatmaps of the learned parameters in Figs. 1 and A3 completely align with the theoretical predictions of Theorem 3.4.

### D.3 VERIFICATION OF THEOREMS 3.5 TO 3.7 AND H.1

We evaluated standardly and adversarially pretrained single-layer transformers on  $\mathcal{D}^{\text{tr}}$ ,  $\mathcal{D}^{\text{te}}$ , the preprocessed MNIST, Fashion-MNIST, and CIFAR-10 datasets. For network parameters, we used the theoretically predicted  $\mathbf{P}^{\text{std}}$  and  $\mathbf{Q}^{\text{std}}$  as standard model parameters and  $\mathbf{P}^{\text{adv}}$  and  $\mathbf{Q}^{\text{adv}}$  as adversarially trained model parameters. This approach allowed us to circumvent the computationally expensive adversarial pretraining for every distinct  $d$  setting. As described previously, our empirical results completely align with the theoretically predicted parameter configurations.

**Configuration in Figs. A4 and A5.** In Fig. A4, the basic settings were  $d = 100$ ,  $\lambda = 0.1$ ,  $N = 1,000$ , and  $\epsilon = 0.15$ . In Fig. A5, they were  $d_{\text{rob}} = 10$ ,  $d_{\text{vul}} = 90$ ,  $d_{\text{irr}} = 0$ ,  $\alpha = 1.0$ ,  $\beta = 0.1$ ,  $\gamma = 0.1$ , and  $\epsilon = 0.2$ . The basic perturbation budget was set to 0.1. We considered 1,000 batches where each batch contained 1,000 in-context demonstrations (i.e.,  $N = 1000$ ), and 1,000 queries. The test distribution  $\mathcal{D}^{\text{te}}$  was constructed based on normal distribution. During sampling,  $yx_i$  was sampled from  $\mathcal{N}(\alpha, \alpha^2)$  for  $i \in \mathcal{S}_{\text{rob}}$ ,  $\mathcal{N}(\beta, \beta^2)$  for  $i \in \mathcal{S}_{\text{vul}}$ , and  $\mathcal{N}(0, \gamma^2)$  for  $i \in \mathcal{S}_{\text{irr}}$ . Each dimension is independent, given  $y$ .

**Configuration in Fig. A6.** The preprocessing procedure is described in Appendix D.1. As batches, we considered 45 binary class pairs from ten classes. The basic perturbation budget was set to 0.1. In the first row of Fig. A6, we used all training samples in the training dataset. As queries, we used all test samples in the test dataset.

**Analysis.** In Figs. A4 to A6, standard transformers consistently demonstrate vulnerability to adversarial attacks, whereas adversarially trained transformers maintain a certain level of robustness, validating Theorems 3.5 and 3.6. However, adversarially pretrained transformers exhibit lower clean accuracy, supporting Theorem 3.7.

In Figs. A4 and A5, we observe that a larger number of vulnerable dimensions increases model vulnerability. Conversely, Fig. A5 shows that a larger number of robust dimensions enhances model robustness. Robust models are less susceptible to increasing vulnerable dimensions and benefit more from increasing robust dimensions.

Additionally, as predicted in Theorems 3.5 and 3.6, standard training exhibits vulnerability to increasing redundant dimensions, which is more detrimental than the harmful effect from increasing vulnerable dimensions, since redundant dimensions do not benefit predictions and are only harmful for robustness. In contrast, adversarially trained transformers exhibit significant resistance to increases in these dimensions.

The second row of Fig. A6 indicates that standard transformers still achieve high classification accuracy in small demonstration regimes, whereas adversarially trained transformers show degraded performance. These results align with our theoretical predictions, Theorem H.1.

## E PROOF OF LEMMA 3.3 AND THEOREM 3.4 (PRETRAINING)

**Lemma 3.3** (Transformation of original optimization problem). The minimization problem (7) can be transformed into the maximization problem  $\max_{\mathbf{b} \in \{0,1\}^{d+1}} \sum_{i=1}^{d+1} \max(0, \sum_{j=1}^{d+1} b_j h_{i,j})$ , where  $h_{i,j} \in \mathbb{R}$  is a constant depending on  $(i, j)$ , and there exists a mapping from  $\mathbf{b}$  to  $\mathbf{P}$  and  $\mathbf{Q}$ .

*Proof.* See ‘‘Overview’’ below in detail. The proof sketch is as follows:

Let us simplify the transformer definition (2) as  $f(x; \theta_1, \theta_2) := \theta_1 x^2 \theta_2 (x + \Delta)$ , where  $\theta_1, \theta_2 \in \{0, 1\}$ . Then the optimization problem (7) becomes

$$\min_{\theta_1, \theta_2 \in \{0,1\}} \max_{\Delta} -y f(x + \Delta; \theta_1, \theta_2) = \min_{\theta_1, \theta_2 \in \{0,1\}} \max_{\Delta} -y(\theta_1 x^2 \theta_2 (x + \Delta)). \quad (\text{A22})$$

This can be transformed to:

$$\min_{\theta_1, \theta_2 \in \{0,1\}} \max_{\Delta} -y(\theta_1 x^2 \theta_2 (x + \Delta)) = \max_{\theta_1, \theta_2 \in \{0,1\}} \min_{\Delta} \theta_1 (y x^2 \theta_2 (x + \Delta)). \quad (\text{A23})$$

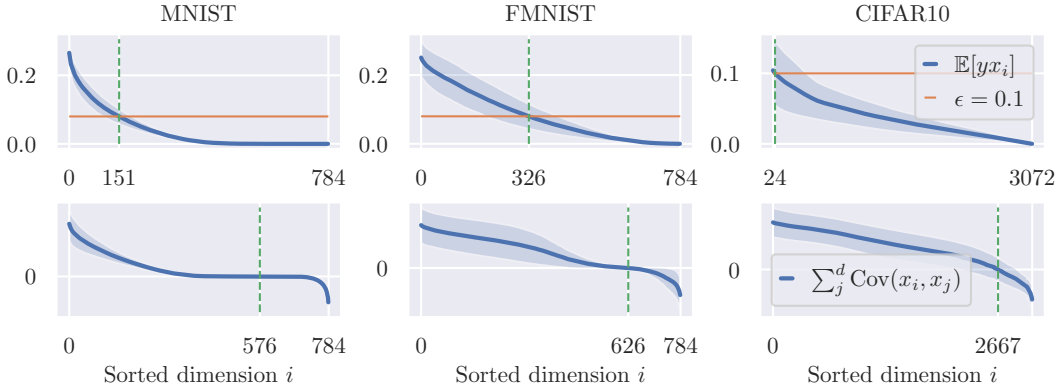


Figure A2: Statistical properties of preprocessed MNIST, Fashion-MNIST, and CIFAR-10 datasets. **First row:** Blue lines represent the mean of  $(1/N) \sum_{n=1}^N y_n \mathbf{x}_n$  across 45 binary class pairs and shaded regions represent the sample standard deviation. Orange lines represent typical perturbation magnitude. Green dashed lines represent the (pseudo) threshold between robust and non-robust dimensions. **Second row:** Blue lines represent the total covariance of each dimension with other dimensions and shaded regions represent sample standard deviation across the 45 binary class pairs. Green dashed lines represent the boundary between positive and negative total covariance.

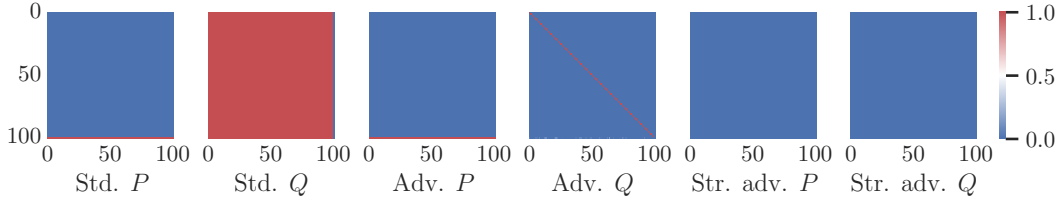


Figure A3: Parameter heatmaps induced by adversarial training (7) with  $d = 100$  and  $\lambda = 0.1$ . For the standard, adversarial, and strong adversarial regimes, we used  $\epsilon = 0$ ,  $\frac{1+(d-1)(\lambda/2)}{d} = 0.06$ , and  $\frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3} = 0.77$ , respectively. We optimized (7) by stochastic gradient descent.

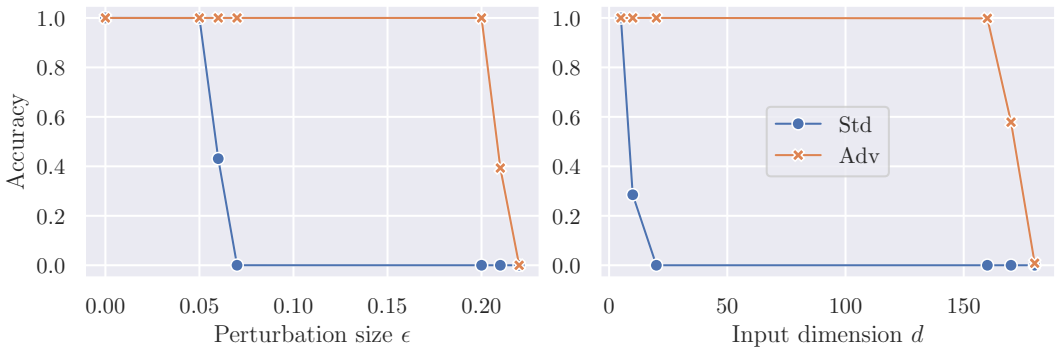


Figure A4: Accuracy (%) of standardly and adversarially pretrained single-layer transformers. Lines represent mean accuracy across batches and shaded regions represent unbiased standard deviation (notably small in magnitude). We used 1,000 batches, each containing 1,000 in-context demonstrations ( $N = 1000$ ) and 1,000 query examples. Base configuration parameters were  $d = 100$ ,  $\lambda = 0.1$ , and  $\epsilon = 0.15$ .

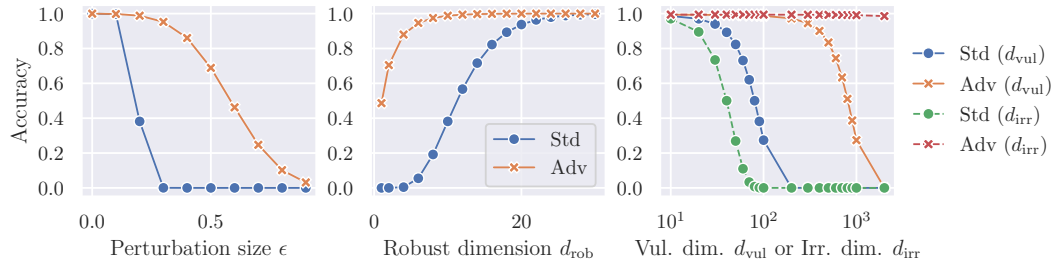


Figure A5: Accuracy (%) of standardly and adversarially pretrained single-layer transformers. Lines represent mean accuracy across batches and shaded regions represent unbiased standard deviation. We used 1,000 batches, each containing 1,000 in-context demonstrations ( $N = 1000$ ) and 1,000 query examples. Base configuration parameters were  $d_{\text{rob}} = 10$ ,  $d_{\text{vul}} = 90$ ,  $d_{\text{irr}} = 0$ ,  $\alpha = 1.0$ ,  $\beta = 0.1$ ,  $\gamma = 0.1$ , and  $\epsilon = 0.2$ .

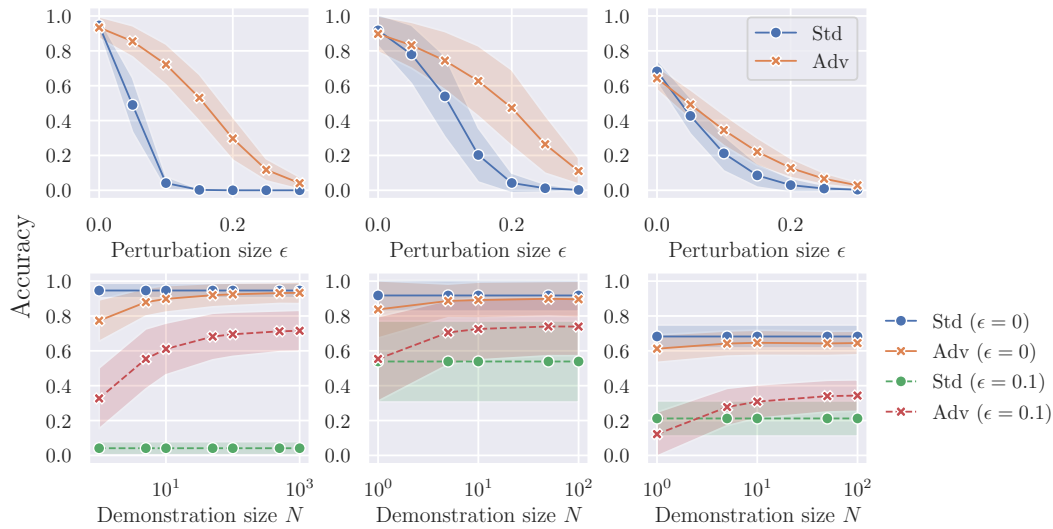


Figure A6: Accuracy (%) of standardly and adversarially pretrained single-layer transformers. Lines represent mean accuracy across 45 binary classification tasks (derived from all possible pairs of the ten classes) and shaded regions represent the unbiased standard deviation. The perturbation size was basically  $\epsilon = 0.1$ .

Since  $\theta_1$  takes only 0 or 1, the optimal strategy is always  $\theta_1 = 1$  when  $yx^2\theta_2(x + \Delta)$  is positive and  $\theta_1 = 0$  when negative. This transforms the problem to:

$$\max_{\theta_1, \theta_2 \in \{0,1\}} \min_{\Delta} \theta_1(yx^2\theta_2(x + \Delta)) = \max_{\theta_2 \in \{0,1\}} \max(0, \min_{\Delta} yx^2\theta_2(x + \Delta)). \quad (\text{A24})$$

Denoting all terms except  $\theta_2$  as  $h$ :

$$\max_{\theta_2 \in \{0,1\}} \max(0, \min_{\Delta} yx^2\theta_2(x + \Delta)) = \max_{\theta_2 \in \{0,1\}} \max(0, \theta_2 h). \quad (\text{A25})$$

This provides intuition for the problem  $\max_{b \in \{0,1\}} \sum_{i=1}^{d(d+1)} \max(0, \sum_{j=1}^{d+1} b_j h_{i,j})$  in [Lemma 3.3](#).  $\square$

**Theorem 3.4** (Parameters learned via adversarial pretraining). The global minimizers of (7) are

$$\begin{aligned} (1. \text{ Standard; } \epsilon = 0) \quad & \mathbf{P} = \mathbf{P}^{\text{std}} := \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{1}_{d+1}^\top \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \mathbf{Q}^{\text{std}} := [\mathbf{1}_{d+1,d} \quad \mathbf{0}_{d+1}]. \\ (2. \text{ Adversarial; } \epsilon = \frac{1+(d-1)(\lambda/2)}{d}) \quad & \mathbf{P} = \mathbf{P}^{\text{adv}} := \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{1}_{d+1}^\top \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \mathbf{Q}^{\text{adv}} := \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix}. \\ (3. \text{ Strongly adversarial; } \epsilon \geq \frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3}) \quad & \mathbf{P} = \mathbf{0}_{d+1,d+1} \quad \text{and} \quad \mathbf{Q} = \mathbf{0}_{d+1,d+1}. \end{aligned}$$

*Proof.* This is the special case of the following theorem.  $\square$

**Theorem E.1** (General case of [Theorem 3.4](#)). The global minimizer of (7) is as follows:

- If

$$0 \leq \epsilon \leq \frac{\lambda(\lambda(d-2) + 4)}{2(\lambda(d-1) + 2)}, \quad (\text{A26})$$

$$\text{then } \mathbf{P} = \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{1}_{d+1}^\top \end{bmatrix} \text{ and } \mathbf{Q} = [\mathbf{1}_{d+1,d} \quad \mathbf{0}_{d+1}].$$

- If

$$\epsilon = \frac{1 + (d-1)(\lambda/2)}{d}, \quad (\text{A27})$$

$$\text{then } \mathbf{P} = \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{1}_{d+1}^\top \end{bmatrix} \text{ and } \mathbf{Q} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix}.$$

- If

$$\epsilon \geq \frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{(d-1)\lambda^2+3}, \quad (\text{A28})$$

$$\text{then } \mathbf{P} = \mathbf{0}_{d+1,d+1} \text{ and } \mathbf{Q} = \mathbf{0}_{d+1,d+1}.$$

*Proof.*

**Overview.** The loss function  $\mathcal{L}(\mathbf{P}, \mathbf{Q})$  is determined only by the last row of  $\mathbf{P}$  and the first  $d$  columns of  $\mathbf{Q}$ . Let

$$\mathbf{P} := \begin{bmatrix} \mathbf{0}_{d,d+1} \\ \mathbf{b}^\top \end{bmatrix}, \quad \mathbf{Q} := [\mathbf{A} \quad \mathbf{0}_{d+1}], \quad (\text{A29})$$

where  $\mathbf{b} \in \mathbb{R}^{d+1}$  and  $\mathbf{A} := [\mathbf{a}_1 \cdots \mathbf{a}_d] \in \mathbb{R}^{(d+1) \times d}$ . With  $\mathbf{b}$ ,  $\mathbf{A}$ , and  $\mathbf{G} := \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top / N$ , the loss function  $\mathcal{L}(\mathbf{P}, \mathbf{Q})$  can be represented as:

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) := \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} \left[ \max_{\|\Delta\|_\infty \leq \epsilon} -y_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}, \mathbf{Q})]_{d+1, N+1} \right] \quad (\text{A30})$$

$$= \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} \left[ \max_{\|\Delta\|_\infty \leq \epsilon} -y_{N+1} \left[ \mathbf{Z}_\Delta + \frac{1}{N} \mathbf{P} \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{Q} \mathbf{Z}_\Delta \right]_{d+1, N+1} \right] \quad (\text{A31})$$

$$= \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} \left[ \max_{\|\Delta\|_\infty \leq \epsilon} -y_{N+1} \mathbf{b}^\top \mathbf{G} \mathbf{A} (\mathbf{x}_{N+1} + \Delta) \right]. \quad (\text{A32})$$

Using  $\mathbf{b}$  and  $\mathbf{A}$ , we redefine the loss function as  $\mathcal{L}(\mathbf{b}, \mathbf{A}) := \mathcal{L}(\mathbf{P}, \mathbf{Q})$ . Since  $\mathbf{G}$  does not include  $\Delta$  and  $\max_{\|\Delta\|_\infty \leq \epsilon} \mathbf{w}^\top \Delta = \epsilon \|\mathbf{w}\|_1$  for  $\mathbf{w} \in \mathbb{R}^d$ , the inner maximization can be solved as:

$$\mathcal{L}(\mathbf{b}, \mathbf{A}) = \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} \left[ -y_{N+1} \mathbf{b}^\top \mathbf{G} \mathbf{A} \mathbf{x}_{N+1} + \epsilon \|\mathbf{b}^\top \mathbf{G} \mathbf{A}\|_1 \right]. \quad (\text{A33})$$

When  $0 \leq \mathbf{b} \leq 1$  and  $0 \leq \mathbf{A} \leq 1$ , then  $\|\mathbf{b}^\top \mathbf{G} \mathbf{A}\|_1 = \mathbf{b}^\top \mathbf{G} \mathbf{A} \mathbf{1}$  since all the elements of  $\mathbf{G}$  are nonnegative. Thus,

$$\begin{aligned} & \min_{0 \leq \mathbf{b} \leq 1, 0 \leq \mathbf{A} \leq 1} \mathcal{L}(\mathbf{b}, \mathbf{A}) \\ &= \min_{0 \leq \mathbf{b} \leq 1, 0 \leq \mathbf{A} \leq 1} \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} \left[ -y_{N+1} \mathbf{b}^\top \mathbf{G} \mathbf{A} \mathbf{x}_{N+1} + \epsilon \mathbf{b}^\top \mathbf{G} \mathbf{A} \mathbf{1} \right]. \end{aligned} \quad (\text{A34})$$

Let the  $i$ -th row of  $\mathbf{G}$  be  $\mathbf{g}_i^\top$ . Rearranging the argument of the expectation as:

$$-y_{N+1} \mathbf{b}^\top \mathbf{G} \mathbf{A} \mathbf{x}_{N+1} + \epsilon \mathbf{b}^\top \mathbf{G} \mathbf{A} \mathbf{1} = - \sum_{j=1}^{d+1} \sum_{k=1}^d A_{j,k} \left( \sum_{i=1}^{d+1} b_i g_{i,j} (y_{N+1} x_{N+1,k} - \epsilon) \right). \quad (\text{A35})$$

Thus, the objective function can be represented as:

$$\max_{0 \leq \mathbf{b} \leq 1, 0 \leq \mathbf{A} \leq 1} \sum_{j=1}^{d+1} \sum_{k=1}^d A_{j,k} \left( \sum_{i=1}^{d+1} b_i \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} [g_{i,j} (y_{N+1} x_{N+1,k} - \epsilon)] \right). \quad (\text{A36})$$

Since the objective function is linear with respect to  $\mathbf{b}$  and  $\mathbf{A}$ , respectively, the optimal solution exists on the boundary:

$$\max_{\mathbf{b} \in \{0,1\}^{d+1}, \mathbf{A} \in \{0,1\}^{(d+1) \times d}} \sum_{j=1}^{d+1} \sum_{k=1}^d A_{j,k} \left( \sum_{i=1}^{d+1} b_i \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} [g_{i,j} (y_{N+1} x_{N+1,k} - \epsilon)] \right). \quad (\text{A37})$$

This is maximized by  $A_{j,k} = 1$  if  $\sum_{i=1}^{d+1} b_i \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} [g_{i,j} (y_{N+1} x_{N+1,k} - \epsilon)] \geq 0$  and 0 otherwise. Now,

$$\max_{\mathbf{b} \in \{0,1\}^{d+1}} \sum_{j=1}^{d+1} \sum_{k=1}^d \phi \left( \sum_{i=1}^{d+1} b_i \mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}} [g_{i,j} (y_{N+1} x_{N+1,k} - \epsilon)] \right), \quad (\text{A38})$$

where  $\phi(x) := \max(0, x)$ . Calculating the expectation and optimizing  $\mathbf{b}$ , we obtain the solution.

**Calculation of the expectation.** First, we consider the expectation given  $c$ . Since  $y_n x_{n,i} = 1$  if  $i = c$  and  $y_n x_{n,i} \sim U(0, \lambda)$  otherwise, the expectation of  $y_n \mathbf{x}_n$  can be calculated as:

$$\mathbb{E}[y_n x_{n,i} | c] = \begin{cases} 1 & (i = c) \\ \frac{\lambda}{2} & (i \neq c) \end{cases}, \quad \mathbb{E}[y_n \mathbf{x}_n^\top | c] = \left[ \frac{\lambda}{2} \quad \cdots \quad \frac{\lambda}{2} \quad \underbrace{1}_{c\text{-th}} \quad \frac{\lambda}{2} \quad \cdots \quad \frac{\lambda}{2} \right]. \quad (\text{A39})$$

The expectation of  $\mathbf{G}$  can be calculated as:

$$\mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{G} | c] = \frac{1}{N} \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top | c] \quad (\text{A40})$$

$$= \frac{1}{N} \left[ \begin{array}{c|c} \sum_{n=1}^N \mathbb{E}_{\mathbf{x}_n} [\mathbf{x}_n \mathbf{x}_n^\top | c] & \sum_{n=1}^N \mathbb{E}_{\mathbf{x}_n, y_n} [y_n \mathbf{x}_n | c] \\ \hline \sum_{n=1}^N \mathbb{E}_{\mathbf{x}_n, y_n} [y_n \mathbf{x}_n^\top | c] & N \end{array} \right] \quad (\text{A41})$$

$$= \left[ \begin{array}{c|c} \mathbb{E}_{\mathbf{x}_n} [\mathbf{x}_n \mathbf{x}_n^\top | c] & \mathbb{E}_{\mathbf{x}_n, y_n} [y_n \mathbf{x}_n | c] \\ \hline \mathbb{E}_{\mathbf{x}_n, y_n} [y_n \mathbf{x}_n^\top | c] & 1 \end{array} \right]. \quad (\text{A42})$$

For  $y_n = 1$  and  $i, j \neq c$ ,  $\mathbb{E}[x_{n,i}^2 | c] = \int_0^\lambda x^2/\lambda dx = \lambda^2/3$  and  $\mathbb{E}[x_{n,i}x_{n,j} | c] = \mathbb{E}[x_{n,i} | c]\mathbb{E}[x_{n,j} | c] = \lambda^2/4$ . Thus,

$$\mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N}[g_{i,j} | c] = \begin{cases} 1 & (i = c) \wedge (j = i, d + 1) \\ \frac{\lambda}{2} & (i = c) \wedge (j \neq i, d + 1) \\ \frac{\lambda^2}{3} & (i \in [d], i \neq c) \wedge (j = i) \\ \frac{\lambda^2}{4} & (i \in [d], i \neq c) \wedge (j = c, d + 1) \\ \frac{\lambda^2}{4} & (i \in [d], i \neq c) \wedge (j \neq i, c, d + 1) \\ 1 & (i = d + 1) \wedge (j = c, d + 1) \\ \frac{\lambda}{2} & (i = d + 1) \wedge (j \neq c, d + 1) \end{cases}. \quad (\text{A43})$$

Note that

$$\mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N}[\mathbf{G} | c] = \begin{matrix} & & & & & \overbrace{\hspace{1.5cm}}^{c\text{-th}} & & & & \\ \begin{bmatrix} \lambda^2/3 & \lambda^2/4 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 \\ \lambda^2/4 & \lambda^2/3 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 \\ \vdots & & & & & & & & & \\ \lambda^2/4 & \lambda^2/4 & \lambda^2/4 & \cdots & \lambda^2/3 & \lambda/2 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 \\ \lambda/2 & \lambda/2 & \lambda/2 & \cdots & \lambda/2 & 1 & \lambda/2 & \cdots & \lambda/2 & 1 \\ \lambda^2/4 & \lambda^2/4 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 & \lambda^2/3 & \cdots & \lambda^2/4 & \lambda/2 \\ \vdots & & & & & & & & & \\ \lambda^2/4 & \lambda^2/4 & \lambda^2/4 & \cdots & \lambda^2/4 & \lambda/2 & \lambda^2/4 & \cdots & \lambda^2/3 & \lambda/2 \\ \lambda/2 & \lambda/2 & \lambda/2 & \cdots & \lambda/2 & 1 & \lambda/2 & \cdots & \lambda/2 & 1 \end{bmatrix} & \end{matrix} \Bigg\} c\text{-th}. \quad (\text{A44})$$

Let

$$h_i(j; k; c) := \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}}[g_{i,j}(y_{N+1}x_{N+1,k} - \epsilon) | c]. \quad (\text{A45})$$

Let  $\epsilon_+ := 1 - \epsilon$  and  $\epsilon_- := \lambda/2 - \epsilon$ . By Eqs. (A39) and (A43),

$$h_i(j; k; c) = \begin{cases} \epsilon_+ & (i \in [d]) \wedge (j = i, d + 1) \wedge (k = i) \wedge (c = i) \\ \epsilon_- & (i \in [d]) \wedge (j = i, d + 1) \wedge (k \neq i) \wedge (c = i) \\ \frac{\lambda}{2}\epsilon_+ & (i \in [d]) \wedge (j \neq i, d + 1) \wedge (k = i) \wedge (c = i) \\ \frac{\lambda}{2}\epsilon_- & (i \in [d]) \wedge (j \neq i, d + 1) \wedge (k \neq i) \wedge (c = i) \\ \frac{\lambda^2}{3}\epsilon_- & (i \in [d]) \wedge (j = i) \wedge (k = i) \wedge (c \neq i) \\ \frac{\lambda}{2}\epsilon_- & (i \in [d]) \wedge (j = c, d + 1) \wedge (k = i) \wedge (c \neq i) \\ \frac{\lambda^2}{4}\epsilon_- & (i \in [d]) \wedge (j \neq i, c, d + 1) \wedge (k = i) \wedge (c \neq i) \\ \frac{\lambda^2}{3}\epsilon_+ & (i \in [d]) \wedge (j = i) \wedge (k = c) \wedge (c \neq i) \\ \frac{\lambda}{2}\epsilon_+ & (i \in [d]) \wedge (j = c, d + 1) \wedge (k = c) \wedge (c \neq i) \\ \frac{\lambda^2}{4}\epsilon_+ & (i \in [d]) \wedge (j \neq i, c, d + 1) \wedge (k = c) \wedge (c \neq i) \\ \frac{\lambda^2}{3}\epsilon_- & (i \in [d]) \wedge (j = i) \wedge (k \neq i, c) \wedge (c \neq i) \\ \frac{\lambda}{2}\epsilon_- & (i \in [d]) \wedge (j = c, d + 1) \wedge (k \neq i, c) \wedge (c \neq i) \\ \frac{\lambda^2}{4}\epsilon_- & (i \in [d]) \wedge (j \neq i, c, d + 1) \wedge (k \neq i, c) \wedge (c \neq i) \\ \epsilon_+ & (i = d + 1) \wedge (j = c, d + 1) \wedge (k = c) \\ \epsilon_- & (i = d + 1) \wedge (j = c, d + 1) \wedge (k \neq c) \\ \frac{\lambda}{2}\epsilon_+ & (i = d + 1) \wedge (j \neq c, d + 1) \wedge (k = c) \\ \frac{\lambda}{2}\epsilon_- & (i = d + 1) \wedge (j \neq c, d + 1) \wedge (k \neq c) \end{cases}. \quad (\text{A46})$$

Then, we compute the expectation along  $c$ . Note that

$$\mathbb{E}_{c, \{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1}}[g_{i,j}(y_{N+1}x_{N+1,k} - \epsilon)] = \frac{1}{d} \sum_{c=1}^d h_i(j; k; c). \quad (\text{A47})$$

Let  $H_{i,j,k} := \sum_{c=1}^d h_i(j; k; c)$ . The summation of  $h_i$  along  $c$  can be calculated as:

For  $(i \in [d]) \wedge (j = i) \wedge (k = i)$ ,

$$H_{i,j,k} = h_i(j = i; k = i; c = i) + \sum_{c \neq i}^d h_i(j = i; k = i; c \neq i) = \epsilon_+ + \frac{\lambda^2}{3}(d-1)\epsilon_- \quad (\text{A48})$$

$$=: r_1. \quad (\text{A49})$$

For  $(i \in [d]) \wedge (j = i) \wedge (k \neq i)$ ,

$$H_{i,j,k} = h_i(j = i; k \neq i; c = i) + h_i(j = i; k = c; c \neq i) + \sum_{c \neq i,k}^d h_i(j = i; k \neq i, c; c \neq i) \quad (\text{A50})$$

$$= \epsilon_- + \frac{\lambda^2}{3}\epsilon_+ + \frac{\lambda^2}{3}(d-2)\epsilon_- \quad (\text{A51})$$

$$=: r_2. \quad (\text{A52})$$

For  $(i \in [d]) \wedge (j = d+1) \wedge (k = i)$ ,

$$H_{i,j,k} = h_i(j = d+1; k = i; c = i) + \sum_{c \neq i}^d h_i(j = d+1; k = i; c \neq i) \quad (\text{A53})$$

$$= \epsilon_+ + \frac{\lambda}{2}(d-1)\epsilon_- \quad (\text{A54})$$

$$=: r_3. \quad (\text{A55})$$

For  $(i \in [d]) \wedge (j = d+1) \wedge (k \neq i)$ ,

$$H_{i,j,k} = h_i(j = d+1; k \neq i; c = i) + h_i(j = d+1; k = c; c \neq i) + \sum_{c \neq i,k}^d h_i(j = d+1; k \neq i, c; c \neq i) \quad (\text{A56})$$

$$= \epsilon_- + \frac{\lambda}{2}\epsilon_+ + \frac{\lambda}{2}(d-2)\epsilon_- \quad (\text{A57})$$

$$=: r_4. \quad (\text{A58})$$

For  $(i \in [d]) \wedge (j \neq i, d+1) \wedge (k = i)$ ,

$$H_{i,j,k} = h_i(j \neq i, d+1; k = i; c = i) + h_i(j = c; k = i; c \neq i) + \sum_{c \neq i,j}^d h_i(j \neq i, c, d+1; k = i; c \neq i) \quad (\text{A59})$$

$$= \frac{\lambda}{2}\epsilon_+ + \frac{\lambda}{2}\epsilon_- + \frac{\lambda^2}{4}(d-2)\epsilon_- \quad (\text{A60})$$

$$=: r_5. \quad (\text{A61})$$

For  $(i \in [d]) \wedge (j \neq i, d+1) \wedge (k \neq i) \wedge (j = k)$ ,

$$H_{i,j,k} = h_i(j \neq i, d+1; k \neq i; c = i) + h_i(j = c; k = c; c \neq i) + \sum_{c \neq i,j,k}^d h_i(j \neq i, c, d+1; k \neq i, c; c \neq i) \quad (\text{A62})$$

$$= \frac{\lambda}{2}\epsilon_- + \frac{\lambda}{2}\epsilon_+ + \frac{\lambda^2}{4}(d-2)\epsilon_- \quad (\text{A63})$$

$$=: r_5. \quad (\text{A64})$$

For  $(i \in [d]) \wedge (j \neq i, d+1) \wedge (k \neq i) \wedge (j \neq k)$ ,

$$H_{i,j,k} = h_i(j \neq i, d+1; k \neq i; c = i) + h_i(j = c; k \neq i, c; c \neq i) + h_i(j \neq i, c, d+1; k = c; c \neq i)$$

$$+ \sum_{c \neq i, j, k}^d h_i(j \neq i, c, d+1; k \neq i, c; c \neq i) \quad (\text{A65})$$

$$= \frac{\lambda}{2}\epsilon_- + \frac{\lambda}{2}\epsilon_- + \frac{\lambda^2}{4}\epsilon_+ + \frac{\lambda^2}{4}(d-3)\epsilon_- \quad (\text{A66})$$

$$=: r_6. \quad (\text{A67})$$

For  $(i = d+1) \wedge (j = d+1)$ ,

$$H_{i,j,k} = h_i(j = d+1; k = c; c = k) + \sum_{c \neq k}^d h_i(j = d+1; k \neq c; c \neq k) \quad (\text{A68})$$

$$= \epsilon_+ + (d-1)\epsilon_- \quad (\text{A69})$$

$$=: r_7. \quad (\text{A70})$$

For  $(i = d+1) \wedge (j \neq d+1) \wedge (j = k)$ ,

$$H_{i,j,k} = h_i(j = c; k = c; c = k) + \sum_{c \neq k}^d h_i(j \neq d+1; k \neq c; c \neq k) \quad (\text{A71})$$

$$= \epsilon_+ + \frac{\lambda}{2}(d-1)\epsilon_- \quad (\text{A72})$$

$$=: r_3. \quad (\text{A73})$$

For  $(i = d+1) \wedge (j \neq d+1) \wedge (j \neq k)$ ,

$$H_{i,j,k} = h_i(j = c; k \neq c; c \neq k) + h_i(j \neq c; k = c; c = k) + \sum_{c \neq j, k}^d h_i(j \neq c, d+1; k \neq c; c \neq k) \quad (\text{A74})$$

$$= \epsilon_- + \frac{\lambda}{2}\epsilon_+ + \frac{\lambda}{2}(d-2)\epsilon_- \quad (\text{A75})$$

$$=: r_4. \quad (\text{A76})$$

**Optimization of  $\mathbf{A}$  and  $\mathbf{b}$ .** From Eq. (A38), we redefine the objective function as:

$$\begin{aligned} & d \max_{\mathbf{b} \in \{0,1\}^{d+1}} \sum_{j=1}^{d+1} \sum_{k=1}^d \phi \left( \sum_{i=1}^{d+1} b_i \mathbb{E}_{c, \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N+1}} [g_{i,j}(y_{N+1} x_{N+1, k} - \epsilon)] \right) \\ &= \max_{\mathbf{b} \in \{0,1\}^{d+1}} \sum_{j=1}^{d+1} \sum_{k=1}^d \phi \left( \sum_{i=1}^{d+1} b_i H_{i,j,k} \right). \end{aligned} \quad (\text{A77})$$

Recall that we set  $A_{j,k} = 1$  if  $\sum_{i=1}^{d+1} b_i H_{i,j,k} \geq 0$  and 0 otherwise. Let  $[d]' := \{i \in [d] \mid b_i = 1\}$  and  $d' := |[d]'$ . Now,

$$\begin{aligned} \sum_{j=1}^{d+1} \sum_{k=1}^d \phi \left( \sum_{i=1}^{d+1} b_i H_{i,j,k} \right) &= \sum_{k=1}^d \phi \left( b_{d+1} H_{d+1, d+1, k} + \mathbb{1}[k \in [d]'] H_{k, d+1, k} + \sum_{i \in [d]', i \neq k} H_{i, d+1, k} \right) \\ &+ \sum_{j=1}^d \phi \left( b_{d+1} H_{d+1, j, j} + \mathbb{1}[j \in [d]'] H_{j, j, j} + \sum_{i \in [d]', i \neq j} H_{i, j, j} \right) \\ &+ \sum_{j=1}^d \sum_{k \neq j}^d \phi \left( b_{d+1} H_{d+1, j, k} + \mathbb{1}[j \in [d]'] H_{i, i, k} \right. \\ &\quad \left. + \mathbb{1}[k \in [d]'] H_{i, j, i} + \sum_{i \in [d]', i \neq j, k} H_{i, j, k} \right). \end{aligned} \quad (\text{A78})$$

By Eqs. (A55), (A58) and (A70),

$$\begin{aligned} & \sum_{k=1}^d \phi \left( b_{d+1} H_{d+1,d+1,k} + \mathbb{1}[k \in [d']] H_{k,d+1,k} + \sum_{i \in [d'], i \neq k} H_{i,d+1,k} \right) \\ &= \sum_{k=1}^d \phi \left( b_{d+1} r_7 + \mathbb{1}[k \in [d']] r_3 + \sum_{i \in [d'], i \neq k} r_4 \right) \end{aligned} \quad (\text{A79})$$

$$= d' \phi \underbrace{(b_{d+1} r_7 + r_3 + (d' - 1) r_4)}_{=: s_1(d', b_{d+1})} + (d - d') \phi \underbrace{(b_{d+1} r_7 + d' r_4)}_{=: s_2(d', b_{d+1})}. \quad (\text{A80})$$

By Eqs. (A49), (A64) and (A73),

$$\begin{aligned} & \sum_{j=1}^d \phi \left( b_{d+1} H_{d+1,j,j} + \mathbb{1}[j \in [d']] H_{j,j,j} + \sum_{i \in [d'], i \neq j} H_{i,j,j} \right) \\ &= \sum_{j=1}^d \phi \left( b_{d+1} r_3 + \mathbb{1}[j \in [d']] r_1 + \sum_{i \in [d'], i \neq j} r_5 \right) \end{aligned} \quad (\text{A81})$$

$$= d' \phi \underbrace{(b_{d+1} r_3 + r_1 + (d' - 1) r_5)}_{=: s_3(d', b_{d+1})} + (d - d') \phi \underbrace{(b_{d+1} r_3 + d' r_5)}_{=: s_4(d', b_{d+1})}. \quad (\text{A82})$$

By Eqs. (A52), (A61), (A67) and (A76),

$$\begin{aligned} & \sum_{j=1}^d \sum_{k \neq j}^d \phi \left( b_{d+1} H_{d+1,j,k} + \mathbb{1}[j \in [d']] H_{i,i,k} + \mathbb{1}[k \in [d']] H_{i,j,i} + \sum_{i \in [d'], i \neq j,k} H_{i,j,k} \right) \\ &= \sum_{j=1}^d \sum_{k \neq j}^d \phi \left( b_{d+1} r_4 + \mathbb{1}[j \in [d']] r_2 + \mathbb{1}[k \in [d']] r_5 + \sum_{i \in [d'], i \neq j,k} r_6 \right) \end{aligned} \quad (\text{A83})$$

$$\begin{aligned} &= d'(d' - 1) \phi \underbrace{(b_{d+1} r_4 + r_2 + r_5 + (d' - 2) r_6)}_{=: s_5(d', b_{d+1})} + d'(d - d') \phi \underbrace{(b_{d+1} r_4 + r_2 + (d' - 1) r_6)}_{=: s_6(d', b_{d+1})} \\ &+ d'(d - d') \phi \underbrace{(b_{d+1} r_4 + r_5 + (d' - 1) r_6)}_{=: s_7(d', b_{d+1})} + (d - d')(d - d' - 1) \phi \underbrace{(b_{d+1} r_4 + d' r_6)}_{=: s_8(d', b_{d+1})}. \end{aligned} \quad (\text{A84})$$

Now,

$$\begin{aligned} \sum_{j=1}^{d+1} \sum_{k=1}^d \phi \left( \sum_{i=1}^{d+1} b_i H_{i,j,k} \right) &= d' \phi(s_1(d', b_{d+1})) + (d - d') \phi(s_2(d', b_{d+1})) + d' \phi(s_3(d', b_{d+1})) \\ &+ (d - d') \phi(s_4(d', b_{d+1})) + d'(d' - 1) \phi(s_5(d', b_{d+1})) \\ &+ d'(d - d') \phi(s_6(d', b_{d+1})) + d'(d - d') \phi(s_7(d', b_{d+1})) \\ &+ (d - d')(d - d' - 1) \phi(s_8(d', b_{d+1})) \end{aligned} \quad (\text{A85})$$

$$=: \text{score}(d', b_{d+1}). \quad (\text{A86})$$

We shall now summarize the discussion to [Lemma E.2](#). The rest of the proof is left to [Lemma E.3](#).  $\square$

### Optimization of transformed problem.

**Lemma E.2.** Let  $\phi(x) := \max(0, x)$ ,  $d \in \mathbb{N}$ ,  $0 < \lambda < 1$ ,  $0 \leq \epsilon < 1$ ,  $\epsilon_+ := 1 - \epsilon$ , and  $\epsilon_- := \lambda/2 - \epsilon$ . In addition, for  $d' \in \{0, \dots, d\}$  and  $b_{d+1} \in \{0, 1\}$ ,

$$r_1 := \epsilon_+ + \frac{\lambda^2}{3}(d - 1)\epsilon_-, \quad (\text{A87})$$

$$r_2 := \epsilon_- + \frac{\lambda^2}{3}\epsilon_+ + \frac{\lambda^2}{3}(d-2)\epsilon_-, \quad (\text{A88})$$

$$r_3 := \epsilon_+ + \frac{\lambda}{2}(d-1)\epsilon_-, \quad (\text{A89})$$

$$r_4 := \epsilon_- + \frac{\lambda}{2}\epsilon_+ + \frac{\lambda}{2}(d-2)\epsilon_-, \quad (\text{A90})$$

$$r_5 := \frac{\lambda}{2}\epsilon_+ + \frac{\lambda}{2}\epsilon_- + \frac{\lambda^2}{4}(d-2)\epsilon_-, \quad (\text{A91})$$

$$r_6 := \frac{\lambda}{2}\epsilon_- + \frac{\lambda}{2}\epsilon_+ + \frac{\lambda^2}{4}\epsilon_+ + \frac{\lambda^2}{4}(d-3)\epsilon_-, \quad (\text{A92})$$

$$r_7 := \epsilon_+ + (d-1)\epsilon_-, \quad (\text{A93})$$

$$s_1(d', b_{d+1}) := b_{d+1}r_7 + r_3 + (d' - 1)r_4, \quad (\text{A94})$$

$$s_2(d', b_{d+1}) := b_{d+1}r_7 + d'r_4, \quad (\text{A95})$$

$$s_3(d', b_{d+1}) := b_{d+1}r_3 + r_1 + (d' - 1)r_5, \quad (\text{A96})$$

$$s_4(d', b_{d+1}) := b_{d+1}r_3 + d'r_5, \quad (\text{A97})$$

$$s_5(d', b_{d+1}) := b_{d+1}r_4 + r_2 + r_5 + (d' - 2)r_6, \quad (\text{A98})$$

$$s_6(d', b_{d+1}) := b_{d+1}r_4 + r_2 + (d' - 1)r_6, \quad (\text{A99})$$

$$s_7(d', b_{d+1}) := b_{d+1}r_4 + r_5 + (d' - 1)r_6, \quad (\text{A100})$$

$$s_8(d', b_{d+1}) := b_{d+1}r_4 + d'r_6, \quad (\text{A101})$$

$$\begin{aligned} \text{score}(d', b_{d+1}) &:= d'\phi(s_1(d', b_{d+1})) + (d - d')\phi(s_2(d', b_{d+1})) + d'\phi(s_3(d', b_{d+1})) \\ &\quad + (d - d')\phi(s_4(d', b_{d+1})) + d'(d' - 1)\phi(s_5(d', b_{d+1})) \\ &\quad + d'(d - d')\phi(s_6(d', b_{d+1})) + d'(d - d')\phi(s_7(d', b_{d+1})) \\ &\quad + (d - d')(d - d' - 1)\phi(s_8(d', b_{d+1})). \end{aligned} \quad (\text{A102})$$

Considering the following optimization problem:

$$\max_{d' \in \{0, \dots, d\}, b_{d+1} \in \{0, 1\}} \text{score}(d', b_{d+1}). \quad (\text{A103})$$

Then, setting  $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{(d+1) \times (d+1)}$  to

$$\mathbf{P} = \begin{bmatrix} \mathbf{0}_{d, d+1} \\ \mathbf{b}^\top \end{bmatrix}, \quad \mathbf{Q} = [\mathbf{A} \quad \mathbf{0}_{d+1}], \quad \mathbf{b}^\top = [\underbrace{1 \quad 1 \quad \dots \quad 1}_{d'} \quad \underbrace{0 \quad 0 \quad \dots \quad 0}_{d-d'} \quad b_{d+1}], \quad (\text{A104})$$

$$A_{jk} = \begin{cases} \mathbb{1}[b_{d+1}r_7 + \mathbb{1}[k \leq d']r_3 + (d' - \mathbb{1}[k \leq d'])r_4 \geq 0] \\ \quad (j = d + 1) \\ \mathbb{1}[b_{d+1}r_3 + \mathbb{1}[j \leq d']r_1 + (d' - \mathbb{1}[j \leq d'])r_5 \geq 0] \\ \quad (j \neq d + 1) \wedge (j = k) \\ \mathbb{1}[b_{d+1}r_4 + \mathbb{1}[j \leq d']r_2 + \mathbb{1}[k \leq d']r_5 + (d' - \mathbb{1}[j \leq d'] - \mathbb{1}[k \leq d'])r_6 \geq 0] \\ \quad (j \neq d + 1) \wedge (j \neq k) \end{cases}, \quad (\text{A105})$$

the global maximizer of (A103) is the global minimizer of (7).

*Proof.* See the above discussion.  $\square$

**Lemma E.3.** The global maximizer of (A103) is as follows:

(a) If

$$0 \leq \epsilon \leq \frac{\lambda(\lambda(d-2)+4)}{2(\lambda(d-1)+2)}, \quad (\text{A106})$$

then  $d' = d$  and  $b_{d+1} = 1$ . This corresponds to  $\mathbf{b} = \mathbf{1}_{d+1}$  and  $\mathbf{A} = \mathbf{1}_{d+1,d}$ .

(b) If

$$\epsilon = \frac{\lambda(d-1)+2}{2d}, \quad (\text{A107})$$

then  $d' = d$  and  $b_{d+1} = 1$ . This corresponds to  $\mathbf{b} = \mathbf{1}_{d+1}$  and  $\mathbf{A} = [\mathbf{I}_d \ \mathbf{0}_d]^\top$ .

(c) If

$$\epsilon \geq \frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{\lambda^2(d-1)+3}, \quad (\text{A108})$$

then  $d' = 0$  and  $b_{d+1} = 0$ . This corresponds to  $\mathbf{b} = \mathbf{1}_{d+1}$  and  $\mathbf{A} = \mathbf{0}_{d+1,d}$ .

*Proof.* For notational simplicity, we abbreviate terms including variables such as  $x_1, x_2, \dots$  (e.g.,  $x_1^2 + 3x_2 + \dots$ ) using the notation  $\Theta(x_1, x_2, \dots)$ . In particular, when the expression is strictly nonnegative (e.g.,  $x_1^2 + x_2^2$ ) or nonpositive, we use  $\Theta_+(x_1, x_2, \dots)$  or  $\Theta_-(x_1, x_2, \dots)$ , respectively. These terms are not essential to the analysis and too long. They can be derived by simple basic arithmetic operations. These concrete values can be showed by our python codes.

We define  $\epsilon_1, \dots, \epsilon_7$  as

$$r_1 = 0 \iff \epsilon = \frac{\lambda}{2} + \frac{3}{2} \frac{2-\lambda}{\lambda^2(d-1)+3} =: \epsilon_1, \quad (\text{A109})$$

$$r_2 = 0 \iff \epsilon = \frac{\lambda(\lambda^2(d-2)+2\lambda+3)}{2(\lambda^2(d-1)+3)} =: \epsilon_2, \quad (\text{A110})$$

$$r_3 = 0 \iff \epsilon = \frac{\lambda^2(d-1)+4}{2(\lambda(d-1)+2)} =: \epsilon_3, \quad (\text{A111})$$

$$r_4 = 0 \iff \epsilon = \frac{\lambda(\lambda(d-2)+4)}{2(\lambda(d-1)+2)} =: \epsilon_4, \quad (\text{A112})$$

$$r_5 = 0 \iff \epsilon = \frac{\lambda^2(d-2)+2\lambda+4}{2(\lambda(d-2)+4)} =: \epsilon_5, \quad (\text{A113})$$

$$r_6 = 0 \iff \epsilon = \frac{\lambda(\lambda(d-3)+6)}{2(\lambda(d-2)+4)} =: \epsilon_6, \quad (\text{A114})$$

$$r_7 = 0 \iff \epsilon = \frac{\lambda(d-1)+2}{2d} =: \epsilon_7, \quad (\text{A115})$$

$$s_5(d, 1) = 0 \iff \epsilon = \frac{\lambda(3d^2\lambda^2 - 8d\lambda^2 + 24d\lambda + 4\lambda^2 - 34\lambda + 48)}{2(3d^2\lambda^2 - 5d\lambda^2 + 18d\lambda + 2\lambda^2 - 18\lambda + 24)} =: \epsilon_{s_5}. \quad (\text{A116})$$

Since

$$\epsilon_1 - \epsilon_3 = \frac{\lambda(d-1)(2-\lambda)(3-2\lambda)}{2(\lambda(d-1)+2)(\lambda^2(d-1)+3)} \geq 0, \quad (\text{A117})$$

$$\epsilon_3 - \epsilon_5 = \frac{(2-\lambda)^2}{(\lambda(d-2)+4)(\lambda(d-1)+2)} \geq 0, \quad (\text{A118})$$

$$\epsilon_5 - \epsilon_7 = \frac{(d-2)(2-\lambda)^2}{2d(\lambda(d-2)+4)} \geq 0, \quad (\text{A119})$$

$$\epsilon_7 - \epsilon_{s_5} = \frac{(2-\lambda)(-3d\lambda^2 + 6d\lambda + 2\lambda^2 - 18\lambda + 24)}{2d(3d^2\lambda^2 - 5d\lambda^2 + 18d\lambda + 2\lambda^2 - 18\lambda + 24)} \geq 0, \quad (\text{A120})$$

$$\epsilon_{s_5} - \epsilon_4 = \frac{\lambda^2(2-\lambda)}{(\lambda(d-1)+2)(3d^2\lambda^2 - 5d\lambda^2 + 18d\lambda + 2\lambda^2 - 18\lambda + 24)} \geq 0, \quad (\text{A121})$$

$$\epsilon_4 - \epsilon_6 = \frac{\lambda(2-\lambda)^2}{2(\lambda(d-2)+4)(\lambda(d-1)+2)} \geq 0, \quad (\text{A122})$$

$$\epsilon_6 - \epsilon_2 = \frac{\lambda(3-\lambda)(2-\lambda)(1-\lambda)}{2(\lambda(d-2)+4)(\lambda^2(d-1)+3)} \geq 0, \quad (\text{A123})$$

for  $d \geq 2$ , they are ordered as

$$\epsilon_2 \leq \epsilon_6 \leq \epsilon_4 \leq \epsilon_{s_5} \leq \epsilon_7 \leq \epsilon_5 \leq \epsilon_3 \leq \epsilon_1. \quad (\text{A124})$$

In score,  $b_{d+1}$  appears as  $b_{d+1}r_3$ ,  $b_{d+1}r_4$ , or  $b_{d+1}r_7$ , each with a positive coefficient in  $d$  and  $d'$ . Thus, if  $r_3, r_4, r_7 \leq 0$ , then  $b_{d+1}$  should be zero. If  $r_3, r_4, r_7 \geq 0$ , then  $b_{d+1}$  should be one. Considering [Ineq. \(A124\)](#), for  $d \geq 2$ , the optimal  $b_{d+1}$  is one if  $\epsilon \leq \epsilon_4$  and zero if  $\epsilon \geq \epsilon_3$ .

**One-Dimensional Case.** If  $d = 1$ ,

$$\begin{aligned} & \text{score}(d', b_{d+1}) \\ &= \mathbb{1}[d' = 0](\phi(b_{d+1}r_7) + \phi(b_{d+1}r_3)) + \mathbb{1}[d' = 1](\phi(b_{d+1}r_7 + r_3) + \phi(b_{d+1}r_3 + r_1)) \end{aligned} \quad (\text{A125})$$

$$\begin{aligned} &= \mathbb{1}[d' = 0](\phi(b_{d+1}\epsilon_+) + \phi(b_{d+1}\epsilon_+)) \\ &+ \mathbb{1}[d' = 1](\phi(b_{d+1}\epsilon_+ + \epsilon_+) + \phi(b_{d+1}\epsilon_+ + \epsilon_+)). \end{aligned} \quad (\text{A126})$$

As  $\epsilon_+$  is always positive for  $0 \leq \epsilon < 1$ ,  $d' = d = 1$  and  $b_{d+1} = 1$  are the optimal. This aligns with the following case analysis.

**Weak Adversarial (Case 1).** Assume  $d \geq 2$  and  $0 \leq \epsilon \leq \epsilon_6$ . As  $\epsilon \leq \epsilon_6 \leq \epsilon_4$ ,  $b_{d+1} = 1$  is the optimal. By [Ineq. \(A124\)](#),  $r_1, r_3, r_4, r_5, r_6, r_7 \geq 0$ . The sign of  $r_2$  depends on  $\epsilon$ . Thus,  $s_1(d', 1), s_2(d', 1), s_3(d', 1), s_4(d', 1), s_7(d', 1), s_8(d', 1) \geq 0$  for  $0 \leq d' \leq d$ . In addition, for  $d' \geq 2$ ,

$$s_5(d', 1) \geq r_4 + r_2 \quad (\text{A127})$$

$$= \frac{\lambda^3}{6}(d-2) + \frac{\lambda^2}{12}(3d-2) + \frac{3\lambda}{2} - \frac{\epsilon}{6}(2\lambda^2(d-1) + 3\lambda(d-1) + 12) \quad (\text{A128})$$

$$\geq \frac{\lambda^2(2-\lambda)(5-2\lambda)}{12(\lambda(d-2)+4)} \quad (\because \epsilon \leq \epsilon_6) \quad (\text{A129})$$

$$\geq 0. \quad (\text{A130})$$

Thus,  $d'(d'-1)s_5(d', 1)$  is nonnegative for  $0 \leq d' \leq d$ . Similarly, by  $s_6(d', 1) \geq r_4 + r_2 \geq 0$  for  $d' \geq 1$ ,  $d'(d'-1)s_6(d', 1)$  is nonnegative for  $0 \leq d' \leq d$ . Thus,

$$\begin{aligned} \text{score}(d', 1) &:= d' s_1(d', 1) + (d-d')s_2(d', 1) + d' s_3(d', 1) + (d-d')s_4(d', 1) \\ &+ d'(d'-1)s_5(d', 1) + d'(d-d')s_6(d', 1) + d'(d-d')s_7(d', 1) \\ &+ (d-d')(d-d'-1)s_8(d', 1) \end{aligned} \quad (\text{A131})$$

$$\begin{aligned} &= dr_7 + d'r_3 + d'(d-1)r_4 + dr_3 + d'r_1 + d'(d-1)r_5 \\ &+ dr_4 + d'r_2 + d'r_5 + d'(d-1)(d-2)r_6. \end{aligned} \quad (\text{A132})$$

This monotonically increases in  $d'$ . Therefore,  $d' = d$  is the optimal. By [Lemma E.2](#),  $\mathbf{b} = \mathbf{1}_{d+1}$ . In addition, from  $s_1(d, 1), s_3(d, 1), s_5(d, 1) \geq 0$ ,  $\mathbf{A} = \mathbf{1}_{d+1, d}$ .

**Weak Adversarial (Case 2).** Assume  $d \geq 2$  and  $\epsilon_6 \leq \epsilon \leq \epsilon_4$ . As  $\epsilon \leq \epsilon_4$ ,  $b_{d+1} = 1$  is the optimal. By [Ineq. \(A124\)](#),  $r_1, r_3, r_4, r_5, r_7 \geq 0$  and  $r_2, r_6 \leq 0$ . Thus,  $s_1(d', 1), s_2(d', 1), s_3(d', 1), s_4(d', 1) \geq 0$ . In addition,

$$s_5(d', 1) \geq s_5(d, 1) \geq \frac{\lambda^2(2-\lambda)}{12(\lambda(d-1)+2)} \geq 0 \quad (\because \epsilon \leq \epsilon_4), \quad (\text{A133})$$

$$s_7(d', 1) \geq s_7(d, 1) \geq \frac{\lambda(2-\lambda)^3}{8(\lambda(d-1)+2)} \geq 0 \quad (\because \epsilon \leq \epsilon_4). \quad (\text{A134})$$

Due to the following inequality,  $s_8(d', 1)$  is always larger than  $s_6(d', 1)$ :

$$s_8(d', 1) - s_6(d', 1) = -\frac{\lambda^3}{24}(d+1) + \frac{5\lambda^2}{12} - \frac{\lambda}{2} + \frac{\epsilon}{12}(\lambda^2(d+2) + 12(1-\lambda)) \quad (\text{A135})$$

$$\geq \frac{\lambda(3-\lambda)(2-\lambda)(1-\lambda)}{6(\lambda(d-2)+4)} \quad (\because \epsilon \geq \epsilon_6) \quad (\text{A136})$$

$$\geq 0. \quad (\text{A137})$$

If  $s_6(d', 1), s_8(d', 1) \geq 0$ ,

$$\frac{d \text{ score}(d', 1)}{dd'} = \frac{(2 + \lambda(d-1) - 2d\epsilon)(\lambda^2(3d^2 - 5d + 2) + 18\lambda(d-1) + 24)}{24} \geq 0. \quad (\text{A138})$$

We used

$$2 + \lambda(d-1) - 2d\epsilon \geq \frac{(2-\lambda)^2}{\lambda(d-1)+2} \geq 0 \quad (\because \epsilon \leq \epsilon_4). \quad (\text{A139})$$

If  $s_6(d', 1) \leq 0, s_8(d', 1) \geq 0$ ,

$$\begin{aligned} \frac{d \text{ score}(d', 1)}{dd'} &= \Theta(d, d', \lambda) - \frac{\epsilon}{12} \{3d\lambda^2((d-d')^2 + 2d'^2) + 6\lambda(2-\lambda) \left\{ \left(d - \frac{1}{2}d'\right)^2 + \frac{11}{4}d'^2 \right\} \right. \\ &\quad \left. + 8dd'\lambda^2 + d'(4\lambda^2 - 36\lambda + 48) \right\} \end{aligned} \quad (\text{A140})$$

$$\begin{aligned} &\geq \Theta(d, \lambda) - \frac{\lambda(2-\lambda)}{24(\lambda(d-1)+2)} d'(9d'\lambda(2-\lambda) + 6\lambda^2(d+1) - 4\lambda(3d+7) + 24) \\ &\quad (\because \epsilon \leq \epsilon_4) \end{aligned} \quad (\text{A141})$$

$$\geq \frac{(2-\lambda)(d\lambda^3 + d\lambda(12-7\lambda) - \lambda^3 + 11\lambda^2 - 30\lambda + 24)}{12(\lambda(d-1)+2)} \quad (\text{A142})$$

$$\geq 0. \quad (\text{A143})$$

We used for  $0 \leq d' \leq d$ ,

$$\begin{aligned} &d'(9d'\lambda(2-\lambda) + 6\lambda^2(d+1) - 4\lambda(3d+7) + 24) \\ &\leq d\lambda(3d\lambda(2-\lambda) + 6\lambda^2 - 28\lambda + 24). \end{aligned} \quad (\text{A144})$$

If  $s_6(d', 1) \leq 0, s_8(d', 1) \leq 0$ ,

$$\begin{aligned} &\frac{d \text{ score}(d', 1)}{dd'} \\ &= \Theta(d, d', \lambda) - \frac{\epsilon}{12} \{3d^2\lambda(\lambda+4) + 6d(-\lambda^2 - \lambda + 2) + 6\lambda + 12(d-1) \\ &\quad + 2d'(3d^2\lambda^2 + 8d\lambda(-\lambda+1) + 4(2\lambda^2 + (d-6)\lambda + 3)) \} \end{aligned} \quad (\text{A145})$$

$$\geq \Theta(d, \lambda) - \frac{\lambda(2-\lambda)}{12(\lambda(d-1)+2)} d'(-3d\lambda^2 + 6d\lambda + 6\lambda^2 - 20\lambda + 12) \quad (\because \epsilon \leq \epsilon_4) \quad (\text{A146})$$

$$\geq \frac{(2-\lambda)(-d\lambda^3 - 8d\lambda^2 + 24d\lambda - 2\lambda^3 + 22\lambda^2 - 60\lambda + 48)}{24(\lambda(d-1)+2)} \quad (\because d' \leq d) \quad (\text{A147})$$

$$\geq 0. \quad (\text{A148})$$

From the above discussion, for any case, ( $s_6, s_8 \geq 0$ ), ( $s_6 \leq 0$  and  $s_8 \geq 0$ ), or ( $s_6, s_8 \leq 0$ ), the derivative of  $\text{score}(d', 1)$  with respect to  $d'$  is nonnegative. Thus,  $d' = d$  is the optimal. By [Lemma E.2](#),  $\mathbf{b} = \mathbf{1}_{d+1}$ . In addition, from  $s_1(d, 1), s_3(d, 1), s_5(d, 1) \geq 0$ ,  $\mathbf{A} = \mathbf{1}_{d+1, d}$ .

**Adversarial.** Assume  $d \geq 2$  and  $\epsilon = \epsilon_7$ . By [Ineq. \(A124\)](#),  $r_1, r_3, r_5 \geq 0$ ,  $r_7 = 0$ , and  $r_2, r_4, r_6 \leq 0$ . Thus,  $s_3(d', b_{d+1}), s_4(d', b_{d+1}) \geq 0$  and  $s_2(d', b_{d+1}), s_6(d', b_{d+1}), s_8(d', b_{d+1}) \leq 0$ . Now,

$$s_1(d', 1) = s_1(d', 0) \geq \frac{(d-d')(2-\lambda)^2}{4d} \geq 0 \quad (\because \epsilon = \epsilon_7). \quad (\text{A149})$$

Thus,

$$\begin{aligned} \text{score}(d', b_{d+1}) &= d' s_1(d', 0) + d' s_3(d', b_{d+1}) + (d-d') s_4(d', b_{d+1}) \\ &\quad + d'(d'-1) \phi(s_5(d', b_{d+1})) + d'(d-d') \phi(s_7(d', b_{d+1})) \\ &= d' s_1(d', 0) + d' r_1 + (d-1) d' r_5 + d b_{d+1} r_3 \end{aligned} \quad (\text{A150})$$

$$\begin{aligned}
& + d'(d' - 1)\phi(b_{d+1}r_4 + r_2 + r_5 + (d' - 2)r_6) \\
& + d'(d - d')\phi(b_{d+1}r_4 + r_5 + (d' - 1)r_6). \tag{A151}
\end{aligned}$$

Since  $r_4$  is nonpositive, this indicates that score changes by  $dr_3 + d'(d - 1)r_4$  at least by switching  $b_{d+1}$  to one from zero. Moreover,

$$dr_3 + d'(d - 1)r_4 \geq \frac{(d - 1)(d - d')(2 - \lambda)^2}{4d} \geq 0 \quad (\because \epsilon = \epsilon_7). \tag{A152}$$

Therefore,  $b_{d+1} = 1$  is the optimal. From [Ineq. \(A124\)](#) and  $\epsilon = \epsilon_7$ ,  $s_7(d', b_{d+1}) - s_5(d', b_{d+1}) \geq 0$ . If  $s_5(d', 1), s_7(d', 1) \geq 0$ ,

$$\frac{d \text{score}(d', 1)}{dd'} = \Theta(d, d', \lambda) - \Theta_+(d, d', \lambda)\epsilon \tag{A153}$$

$$= \Theta(d, \lambda) - \Theta_+(d, \lambda)d' \quad (\because \epsilon = \epsilon_7) \tag{A154}$$

$$\geq 0 \quad (\because d' \leq d_{s_5}), \tag{A155}$$

where

$$s_5(d', 1) \geq 0 \iff d' \leq \frac{3d\lambda^2 - 6d\lambda + 2\lambda^2 - 18\lambda + 24}{6\lambda(\lambda - 2)} =: d_{s_5}. \tag{A156}$$

When  $s_5(d', 1) \leq 0, s_7(d', 1) \geq 0$ , then  $\frac{d \text{score}(d', 1)}{dd'} \geq 0$  similarly holds. If  $s_5(d', 1), s_7(d', 1) \leq 0$ ,  $\frac{d \text{score}(d', 1)}{dd'} \geq 0$  for  $d' \leq d - 1$ . Comparing  $\text{score}(d', 1)$  with  $d' = d - 1$  and  $d' = d$ , we obtain  $\text{score}(d, 1) \geq \text{score}(d - 1, 1)$ . In summary,  $d' = d$  is the optimal. By [Lemma E.2](#),  $\mathbf{b} = \mathbf{1}_{d+1}$ . In addition, from  $s_3(d, 1) \geq 0, s_1(d, 1) = 0$ , and  $s_5(d, 1) < 0$ ,  $\mathbf{A} = [\mathbf{I}_d \ \mathbf{0}_d]^\top$ .

**Strong Adversarial.** Assume  $d \geq 2$  and  $\epsilon \geq \epsilon_1$ . By [Ineq. \(A124\)](#),  $r_1, \dots, r_7$  are nonpositive. Thus,  $s_1(d', b_{d+1}), \dots, s_8(d', b_{d+1})$  are nonpositive. Therefore,  $d' = 0$  and  $b_{d+1} = 0$  are the optimal. By [Lemma E.2](#),  $\mathbf{b} = \mathbf{0}_{d+1}$  and  $\mathbf{A} = \mathbf{0}_{d+1, d}$ .  $\square$

## F PROOF OF [THEOREMS 3.5](#) AND [3.6](#) (ROBUSTNESS)

For notational convenience, we occasionally describe representations and equations under the assumption that  $\mathcal{S}_{\text{rob}} := \{1, \dots, d_{\text{rob}}\}$ ,  $\mathcal{S}_{\text{vul}} := \{d_{\text{rob}} + 1, \dots, d_{\text{rob}} + d_{\text{vul}}\}$ , and  $\mathcal{S}_{\text{irr}} := \{d_{\text{rob}} + d_{\text{vul}} + 1, \dots, d_{\text{rob}} + d_{\text{vul}} + d_{\text{irr}}\}$ . This assumption is made without loss of generality.

We use *uniform* big-O and -Theta notation. Denote  $f(x) = \mathcal{O}(g(x))$  if there exists a positive constant  $C > 0$  such that  $|f(x)| \leq C|g(x)|$  for every  $x$  in the domain. Denote  $f(x) = \Theta(g(x))$  if there exist  $C_1, C_2 > 0$  such that  $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$  for every  $x$  in the domain.

For notational simplicity, we abbreviate the following matrix:

$$\begin{bmatrix} C_1\alpha \\ C_2\alpha \\ \vdots \\ C_{d_{\text{rob}}}\alpha \\ C_{d_{\text{rob}}+1}\beta \\ \vdots \\ C_{d_{\text{rob}}+d_{\text{vul}}}\beta \\ C_{d_{\text{rob}}+d_{\text{vul}}+1}\gamma \\ \vdots \\ C_{d_{\text{rob}}+d_{\text{vul}}+d_{\text{irr}}}\gamma \end{bmatrix} \quad \text{as} \quad \begin{bmatrix} C_i\alpha \\ C_i\beta \\ C_i\gamma \end{bmatrix}. \tag{A157}$$

**Theorem 3.5** (Standard pretraining case). There exist a constant  $C > 0$  and a strictly positive function  $g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma)$  such that

$$\mathbb{E}_{\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N+1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}^{\text{te}}} \left[ \min_{\|\Delta\|_\infty \leq \epsilon} \mathbf{y}_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}^{\text{std}}, \mathbf{Q}^{\text{std}})]_{d+1, N+1} \right]$$

$$\leq g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) \left\{ \underbrace{C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta)}_{\text{Prediction for original data}} - \underbrace{(d_{\text{rob}} + d_{\text{vul}} + d_{\text{irr}})\epsilon}_{\text{Adversarial effect}} \right\}. \quad (8)$$

*Proof.* Since  $\mathbf{b} = \mathbf{1}_{d+1}$ ,  $\mathbf{A} = \mathbf{1}_{d+1,d}$ , and  $\mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top$  is positive semidefinite, every entry in  $\mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A}$  is nonnegative. Thus, we can solve the inner minimization as

$$\min_{\|\Delta\|_\infty \leq \epsilon} y_{N+1} [\mathbf{f}(\mathbf{Z}_\Delta; \mathbf{P}, \mathbf{Q})]_{d+1, N+1} = \min_{\|\Delta\|_\infty \leq \epsilon} \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} y_{N+1} (\mathbf{x}_{N+1} + \Delta) \quad (\text{A158})$$

$$= \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} (y_{N+1} \mathbf{x}_{N+1} - \epsilon \mathbf{1}_d). \quad (\text{A159})$$

Using  $(\mathbf{x}, y) \sim \mathcal{D}^{\text{te}}$ ,

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{N} \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \right] \\ &= \begin{bmatrix} \mathbb{E}[\mathbf{x}\mathbf{x}^\top] & \mathbb{E}[y\mathbf{x}] \\ \mathbb{E}[y\mathbf{x}^\top] & 1 \end{bmatrix} \end{aligned} \quad (\text{A160})$$

$$= \begin{bmatrix} \mathbb{E}[y\mathbf{x}]\mathbb{E}[y\mathbf{x}^\top] & \mathbb{E}[y\mathbf{x}] \\ \mathbb{E}[y\mathbf{x}^\top] & 1 \end{bmatrix} + \begin{bmatrix} \mathbb{E}[(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])^\top] & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix}. \quad (\text{A161})$$

Since the second term is positive semidefinite,

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{N} \mathbf{1}_{d+1}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{1}_{d+1} \right] \\ &= \mathbf{1}_{d+1}^\top \left( \begin{bmatrix} \mathbb{E}[y\mathbf{x}]\mathbb{E}[y\mathbf{x}^\top] & \mathbb{E}[y\mathbf{x}] \\ \mathbb{E}[y\mathbf{x}^\top] & 1 \end{bmatrix} + \begin{bmatrix} \mathbb{E}[(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])^\top] & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix} \right) \mathbf{1}_{d+1} \end{aligned} \quad (\text{A162})$$

$$\geq \mathbf{1}_{d+1}^\top \begin{bmatrix} \mathbb{E}[y\mathbf{x}^\top]\mathbb{E}[y\mathbf{x}] & \mathbb{E}[y\mathbf{x}] \\ \mathbb{E}[y\mathbf{x}^\top] & 1 \end{bmatrix} \mathbf{1}_{d+1}. \quad (\text{A163})$$

Since every entry of  $\mathbb{E}[y\mathbf{x}^\top]\mathbb{E}[y\mathbf{x}]$  and  $\mathbb{E}[y\mathbf{x}]$  is nonnegative,

$$\mathbb{E} \left[ \frac{1}{N} \mathbf{1}_{d+1}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{1}_{d+1} \right] \geq \mathbf{1}_{d+1}^\top \begin{bmatrix} \mathbb{E}[y\mathbf{x}^\top]\mathbb{E}[y\mathbf{x}] & \mathbb{E}[y\mathbf{x}] \\ \mathbb{E}[y\mathbf{x}^\top] & 1 \end{bmatrix} \mathbf{1}_{d+1} \geq 1. \quad (\text{A164})$$

Representing  $\mathbb{E}[\mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} / N] = [g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) \cdots g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma)]$  using some positive function  $g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) > 0$ , there exists a positive constant  $C > 0$  such that

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} (y_{N+1} \mathbf{x}_{N+1} - \epsilon \mathbf{1}_d) \right] \\ &= \begin{bmatrix} g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) \\ \vdots \\ g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) \end{bmatrix}^\top (\mathbb{E}[y_{N+1} \mathbf{x}_{N+1}] - \epsilon \mathbf{1}_d) \end{aligned} \quad (\text{A165})$$

$$= g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) (\Theta(d_{\text{rob}}\alpha + d_{\text{vul}}\beta) - d\epsilon) \quad (\text{A166})$$

$$\leq g(d_{\text{rob}}, d_{\text{vul}}, d_{\text{irr}}, \alpha, \beta, \gamma) (C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta) - (d_{\text{rob}} + d_{\text{vul}} + d_{\text{irr}})\epsilon). \quad (\text{A167})$$

□

**Theorem 3.6** (Adversarial pretraining case). Suppose that  $q_{\text{rob}}$  and  $q_{\text{vul}}$  defined in [Assumption 3.2](#) are sufficiently small. There exist constants  $C_1, C_2 > 0$  such that

$$\begin{aligned} & \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1} \sim \mathcal{D}^{\text{te}}} \left[ \min_{\|\Delta\|_\infty \leq \epsilon} y_{N+1} [\mathbf{f}(\mathbf{Z}_\Delta; \mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}})]_{d+1, N+1} \right] \\ & \geq \underbrace{C_1(d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1)(d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2)}_{\text{Prediction for original data}} \end{aligned}$$

$$- C_2 \underbrace{\left\{ (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \left( d_{\text{rob}}\alpha + d_{\text{vul}}\beta + \frac{d_{\text{irr}}\gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right\}}_{\text{Adversarial effect}} \epsilon. \quad (9)$$

*Proof.* This is the special case of the following theorem.  $\square$

**Theorem F.1** (General case of [Theorem 3.6](#)). There exist constants  $C, C', C'' > 0$  such that

$$\begin{aligned} & \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^{N+1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}^{\text{te}}} \left[ \min_{\|\Delta\|_\infty \leq \epsilon} y_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}})]_{d+1, N+1} \right] \\ & \geq C(d_{\text{rob}}\alpha + d_{\text{vul}}\beta) \{ (1 - cq_{\text{rob}})d_{\text{rob}}\alpha^2 + (1 - cq_{\text{vul}})d_{\text{vul}}\beta^2 \} + C'(d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2) \\ & \quad - C'' \left\{ (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \left( d_{\text{rob}}\alpha + d_{\text{vul}}\beta + \frac{d_{\text{irr}}\gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right\} \epsilon, \end{aligned} \quad (\text{A168})$$

where

$$c := \frac{(\max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i) (\max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_{i,2})}{\min_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i^3}. \quad (\text{A169})$$

In particular, if there exists a constant  $C''' > 0$  such that  $1 - cq_{\text{rob}} \geq C'''$  and  $1 - cq_{\text{vul}} \geq C'''$ , then there exist constants  $C_1, C_2 > 0$  such that [Ineq. \(9\)](#) holds.

*Proof.* Similarly to [Eq. \(A33\)](#), we can solve the minimization as

$$\begin{aligned} & \min_{\|\Delta\|_\infty \leq \epsilon} y_{N+1} [f(\mathbf{Z}_\Delta; \mathbf{P}, \mathbf{Q})]_{d+1, N+1} \\ & = \min_{\|\Delta\|_\infty \leq \epsilon} \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} y_{N+1} (\mathbf{x}_{N+1} + \Delta) \end{aligned} \quad (\text{A170})$$

$$= \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} y_{N+1} \mathbf{x}_{N+1} - \epsilon \left\| \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} \right\|_1. \quad (\text{A171})$$

By [Eq. \(A161\)](#), we can rearrange the first term as

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} y_{N+1} \mathbf{x}_{N+1} \right] \\ & = \mathbf{1}_{d+1}^\top \begin{bmatrix} \mathbb{E}[y\mathbf{x}] \mathbb{E}[y\mathbf{x}^\top] \\ \mathbb{E}[y\mathbf{x}^\top] \end{bmatrix} \mathbb{E}[y_{N+1} \mathbf{x}_{N+1}] + \mathbf{1}_d^\top \mathbb{E}[(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])^\top] \mathbb{E}[y_{N+1} \mathbf{x}_{N+1}]. \end{aligned} \quad (\text{A172})$$

The first term of [Eq. \(A172\)](#) can be rearranged as

$$\begin{aligned} & \mathbf{1}_{d+1}^\top \begin{bmatrix} \mathbb{E}[y\mathbf{x}] \mathbb{E}[y\mathbf{x}^\top] \\ \mathbb{E}[y\mathbf{x}^\top] \end{bmatrix} \mathbb{E}[y_{N+1} \mathbf{x}_{N+1}] \\ & = \mathbf{1}_{d+1}^\top \begin{bmatrix} C_i C_j \alpha^2 & C_i C_j \alpha \beta & \mathbf{0} \\ C_i C_j \alpha \beta & C_i C_j \beta^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & C_i^2 \gamma^2 \mathbf{I} \\ C_i \alpha & C_i \beta & \mathbf{0} \end{bmatrix} \begin{bmatrix} C_i \alpha \\ C_i \beta \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (\text{A173})$$

$$= \left( \sum_{i \in \mathcal{S}_{\text{rob}}} C_i \alpha + \sum_{i \in \mathcal{S}_{\text{vul}}} C_i \beta + 1 \right) \left( \sum_{i \in \mathcal{S}_{\text{rob}}} C_i^2 \alpha^2 + \sum_{i \in \mathcal{S}_{\text{vul}}} C_i^2 \beta^2 \right) \quad (\text{A174})$$

$$= \left( \min_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i^3 \right) (d_{\text{rob}}\alpha + d_{\text{vul}}\beta) (d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2) + \sum_{i \in \mathcal{S}_{\text{rob}}} C_i^2 \alpha^2 + \sum_{i \in \mathcal{S}_{\text{vul}}} C_i^2 \beta^2. \quad (\text{A175})$$

Consider the second term of [Eq. \(A172\)](#). Now,

$$|\mathbb{E}[(y x_i - \mathbb{E}[y x_i])(y x_j - \mathbb{E}[y x_j])]|$$

$$\leq \begin{cases} \sqrt{C_{i,2}}\sqrt{C_{j,2}}\alpha^2 & (i, j \in \mathcal{S}_{\text{rob}}) \\ \sqrt{C_{i,2}}\sqrt{C_{j,2}}\beta^2 & (i, j \in \mathcal{S}_{\text{vul}}) \\ \sqrt{C_{i,2}}\sqrt{C_{j,2}}\alpha\beta & (i \in \mathcal{S}_{\text{rob}} \wedge j \in \mathcal{S}_{\text{vul}}) \vee (i \in \mathcal{S}_{\text{vul}} \wedge j \in \mathcal{S}_{\text{rob}}) \end{cases}. \quad (\text{A176})$$

Let

$$\mathcal{S} := \left\{ i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}} \mid \sum_{j \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} \mathbb{E}[(yx_i - \mathbb{E}[yx_i])(yx_j - \mathbb{E}[yx_j])] < 0 \right\}. \quad (\text{A177})$$

The second term of Eq. (A172) can be computed as

$$\begin{aligned} & \mathbf{1}_d^\top \mathbb{E}[(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])(y\mathbf{x} - \mathbb{E}[y\mathbf{x}])^\top] \mathbb{E}[y_{N+1}\mathbf{x}_{N+1}] \\ & \geq - \left[ \begin{array}{c} \sqrt{C_{i,2}}\alpha \left( \sum_{j \in \mathcal{S}_{\text{rob}}} \sqrt{C_{j,2}}\alpha + \sum_{j \in \mathcal{S}_{\text{vul}}} \sqrt{C_{j,2}}\beta \right) \\ \vdots \\ \sqrt{C_{i,2}}\alpha \left( \sum_{j \in \mathcal{S}_{\text{rob}}} \sqrt{C_{j,2}}\alpha + \sum_{j \in \mathcal{S}_{\text{vul}}} \sqrt{C_{j,2}}\beta \right) \\ \mathbf{0} \\ \sqrt{C_{i,2}}\beta \left( \sum_{j \in \mathcal{S}_{\text{rob}}} \sqrt{C_{j,2}}\alpha + \sum_{j \in \mathcal{S}_{\text{vul}}} \sqrt{C_{j,2}}\beta \right) \\ \vdots \\ \sqrt{C_{i,2}}\beta \left( \sum_{j \in \mathcal{S}_{\text{rob}}} \sqrt{C_{j,2}}\alpha + \sum_{j \in \mathcal{S}_{\text{vul}}} \sqrt{C_{j,2}}\beta \right) \\ \mathbf{0} \end{array} \right]^\top \begin{bmatrix} C_i\alpha \\ C_i\beta \\ \mathbf{0} \end{bmatrix} \quad (\text{A178}) \\ & = - \left( \sum_{i \in \mathcal{S}_{\text{rob}}} \sqrt{C_{i,2}}\alpha + \sum_{i \in \mathcal{S}_{\text{vul}}} \sqrt{C_{i,2}}\beta \right) \end{aligned}$$

$$\times \left( \sum_{i \in \mathcal{S}_{\text{rob}} \cap \mathcal{S}} C_i \sqrt{C_{i,2}}\alpha^2 + \sum_{i \in \mathcal{S}_{\text{vul}} \cap \mathcal{S}} C_i \sqrt{C_{i,2}}\beta^2 \right) \quad (\text{A179})$$

$$\geq - \left( \max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} \sqrt{C_{i,2}} \right) \left( \max_{i \in (\mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}) \cap \mathcal{S}} C_i \sqrt{C_{i,2}} \right) \times (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)(q_{\text{rob}}d_{\text{rob}}\alpha^2 + q_{\text{vul}}d_{\text{vul}}\beta^2) \quad (\text{A180})$$

$$\geq - \left( \max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i \right) \left( \max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_{i,2} \right) (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)(q_{\text{rob}}d_{\text{rob}}\alpha^2 + q_{\text{vul}}d_{\text{vul}}\beta^2). \quad (\text{A181})$$

By Lemma F.2, we can compute the second term as

$$\begin{aligned} & \mathbb{E} \left[ \left\| \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} \right\|_1 \right] \\ & = \mathcal{O} \left( (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \left( d_{\text{rob}}\alpha + d_{\text{vul}}\beta + \frac{d_{\text{irr}}\gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right). \quad (\text{A182}) \end{aligned}$$

Finally,

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} y_{N+1} \mathbf{x}_{N+1} \right] - \epsilon \mathbb{E} \left[ \left\| \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} \right\|_1 \right] \\ & \geq \left( \min_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i^3 \right) (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)(d_{\text{rob}}\alpha^2 + d_{\text{vul}}\beta^2) + \sum_{i \in \mathcal{S}_{\text{rob}}} C_i^2 \alpha^2 + \sum_{i \in \mathcal{S}_{\text{vul}}} C_i^2 \beta^2 \\ & \quad - \left( \max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_i \right) \left( \max_{i \in \mathcal{S}_{\text{rob}} \cup \mathcal{S}_{\text{vul}}} C_{i,2} \right) (d_{\text{rob}}\alpha + d_{\text{vul}}\beta)(q_{\text{rob}}d_{\text{rob}}\alpha^2 + q_{\text{vul}}d_{\text{vul}}\beta^2) \\ & \quad + \mathcal{O} \left( (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \left( d_{\text{rob}}\alpha + d_{\text{vul}}\beta + \frac{d_{\text{irr}}\gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right). \quad (\text{A183}) \end{aligned}$$

□

**Lemma F.2.** If  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$  are i.i.d. and follow  $\mathcal{D}^{\text{te}}$ , then

$$\begin{aligned} & \mathbb{E} \left[ \left\| \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} \right\|_1 \right] \\ &= \mathcal{O} \left( (d_{\text{rob}} \alpha + d_{\text{vul}} \beta + 1) \left( d_{\text{rob}} \alpha + d_{\text{vul}} \beta + \frac{d_{\text{irr}} \gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right), \end{aligned} \quad (\text{A184})$$

where  $\mathbf{b} = \mathbf{1}_{d+1}$  and  $\mathbf{A}^\top := [\mathbf{I}_d \quad \mathbf{0}_d]$ .

*Proof.* We can rearrange the given expectation as

$$\mathbb{E} \left[ \left\| \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_\Delta \mathbf{M} \mathbf{Z}_\Delta^\top \mathbf{A} \right\|_1 \right] = \mathbb{E} \left[ \left\| \frac{1}{N} \mathbf{1}_{d+1}^\top \begin{bmatrix} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top & \sum_{n=1}^N y_n \mathbf{x}_n \\ \sum_{n=1}^N y_n \mathbf{x}_n^\top & N \end{bmatrix} \begin{bmatrix} \mathbf{I}_d \\ \mathbf{0}_d^\top \end{bmatrix} \right\|_1 \right] \quad (\text{A185})$$

$$= \mathbb{E} \left[ \left\| \frac{1}{N} \mathbf{1}_{d+1}^\top \begin{bmatrix} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top \\ \sum_{n=1}^N y_n \mathbf{x}_n^\top \end{bmatrix} \right\|_1 \right] \quad (\text{A186})$$

$$= \sum_{i=1}^d \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{n=1}^N \left( y_n + \sum_{j=1}^d x_{n,j} \right) x_{n,i} \right\| \right]. \quad (\text{A187})$$

By the Lyapunov inequality, for  $N + 1$  i.i.d. random variables  $X, X_1, \dots, X_N$ ,

$$\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{n=1}^N X_n \right\| \right] \leq \sqrt{\mathbb{E} \left[ \left( \frac{1}{N} \sum_{n=1}^N X_n \right)^2 \right]} = \sqrt{\frac{1}{N} \mathbb{E}[X^2] + \frac{N-1}{N} \mathbb{E}[X]^2}. \quad (\text{A188})$$

Thus, using  $(\mathbf{x}, y) \sim \mathcal{D}^{\text{te}}$ ,

$$\begin{aligned} & \sum_{i=1}^d \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{n=1}^N \left( y_n + \sum_{j=1}^d x_{n,j} \right) x_{n,i} \right\| \right] \\ & \leq \sum_{i=1}^d \sqrt{\frac{1}{N} \mathbb{E} \left[ \left( y + \sum_{j=1}^d x_j \right)^2 x_i^2 \right] + \frac{N-1}{N} \mathbb{E} \left[ \left( y + \sum_{j=1}^d x_j \right)^2 x_i \right]^2}. \end{aligned} \quad (\text{A189})$$

From [Lemma F.3](#), we can compute the second term of using

$$\mathbb{E} \left[ \left( y + \sum_{j=1}^d x_j \right) x_i \right] = \mathbb{E}[y x_i] + \sum_{j=1}^d \mathbb{E}[x_j x_i] \quad (\text{A190})$$

$$= \begin{cases} \mathcal{O}(\alpha(d_{\text{rob}} \alpha + d_{\text{vul}} \beta + 1)) & (i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta(d_{\text{rob}} \alpha + d_{\text{vul}} \beta + 1)) & (i \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^2) & (i \in \mathcal{S}_{\text{irr}}) \end{cases}. \quad (\text{A191})$$

From [Lemma F.3](#), we can compute the first term of using

$$\mathbb{E} \left[ \left( y + \sum_{j=1}^d x_j \right)^2 x_i^2 \right] = \mathbb{E}[x_i^2] + 2 \sum_{j=1}^d \mathbb{E}[y x_j x_i^2] + \sum_{j,k=1}^d \mathbb{E}[x_j x_k x_i^2] \quad (\text{A192})$$

$$= \begin{cases} \mathcal{O}(\alpha^2 \{ (d_{\text{rob}} \alpha + d_{\text{vul}} \beta + 1)^2 + d_{\text{irr}} \gamma^2 \}) & (i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2 \{ (d_{\text{rob}} \alpha + d_{\text{vul}} \beta + 1)^2 + d_{\text{irr}} \gamma^2 \}) & (i \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^2 \{ (d_{\text{rob}} \alpha + d_{\text{vul}} \beta + 1)^2 + d_{\text{irr}} \gamma^2 \}) & (i \in \mathcal{S}_{\text{irr}}) \end{cases}. \quad (\text{A193})$$

Thus,

$$\begin{aligned}
& \sum_{i=1}^d \sqrt{\frac{1}{N} \mathbb{E} \left[ \left( y + \sum_{j=1}^d x_j \right)^2 x_i^2 \right] + \frac{N-1}{N} \mathbb{E} \left[ \left( y + \sum_{j=1}^d x_j \right) x_i \right]^2} \\
&= \mathcal{O} \left( d_{\text{rob}} \left( \alpha(d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) + \sqrt{\frac{d_{\text{irr}}}{N}} \alpha\gamma \right) \right. \\
&\quad \left. + d_{\text{vul}} \left( \beta(d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) + \sqrt{\frac{d_{\text{irr}}}{N}} \beta\gamma \right) \right. \\
&\quad \left. + d_{\text{irr}} \left( \gamma^2 + \frac{\gamma}{\sqrt{N}} \left( (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) + \sqrt{d_{\text{irr}}}\gamma \right) \right) \right) \tag{A194}
\end{aligned}$$

$$= \mathcal{O} \left( (d_{\text{rob}}\alpha + d_{\text{vul}}\beta + 1) \left( d_{\text{rob}}\alpha + d_{\text{vul}}\beta + \frac{d_{\text{irr}}\gamma}{\sqrt{N}} \right) + d_{\text{irr}} \left( \sqrt{\frac{d_{\text{irr}}}{N}} + 1 \right) \gamma^2 \right). \tag{A195}$$

□

**Lemma F.3.** If  $(x, y) \sim \mathcal{D}^{\text{te}}$ , then

(a)

$$\mathbb{E}[x_j x_i] = \begin{cases} \mathcal{O}(\alpha^2) & (i, j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2) & (i, j \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^2) & (i = j) \wedge (i, j \in \mathcal{S}_{\text{irr}}) \\ \mathcal{O}(\alpha\beta) & (i \in \mathcal{S}_{\text{rob}} \wedge j \in \mathcal{S}_{\text{vul}}) \vee (i \in \mathcal{S}_{\text{vul}} \wedge j \in \mathcal{S}_{\text{rob}}) \\ 0 & (i \neq j) \wedge (i \in \mathcal{S}_{\text{irr}} \vee j \in \mathcal{S}_{\text{irr}}) \end{cases}. \tag{A196}$$

(b)

$$\mathbb{E}[y x_j x_i^2] = \begin{cases} \mathcal{O}(\alpha^3) & (i, j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^3) & (i, j \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha^2\beta) & (i \in \mathcal{S}_{\text{rob}} \wedge j \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha\beta^2) & (i \in \mathcal{S}_{\text{vul}} \wedge j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\alpha\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j \in \mathcal{S}_{\text{vul}}) \\ 0 & (j \in \mathcal{S}_{\text{irr}}) \end{cases}. \tag{A197}$$

(c)

$$\begin{aligned}
& \mathbb{E}[x_j x_k x_i^2] \\
&= \begin{cases} \mathcal{O}(\alpha^4) & (i, j, k \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^4) & (i, j, k \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^4) & (j = k) \wedge (i, j, k \in \mathcal{S}_{\text{irr}}) \\ \mathcal{O}(\alpha^3\beta) & (i \in \mathcal{S}_{\text{rob}}) \wedge \{(j \in \mathcal{S}_{\text{rob}} \wedge k \in \mathcal{S}_{\text{vul}}) \vee (j \in \mathcal{S}_{\text{vul}} \wedge k \in \mathcal{S}_{\text{rob}})\} \\ \mathcal{O}(\alpha\beta^3) & (i \in \mathcal{S}_{\text{vul}}) \wedge \{(j \in \mathcal{S}_{\text{rob}} \wedge k \in \mathcal{S}_{\text{vul}}) \vee (j \in \mathcal{S}_{\text{vul}} \wedge k \in \mathcal{S}_{\text{rob}})\} \\ \mathcal{O}(\alpha^2\beta^2) & (i \in \mathcal{S}_{\text{rob}} \wedge j, k \in \mathcal{S}_{\text{vul}}) \vee (i \in \mathcal{S}_{\text{vul}} \wedge j, k \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\alpha^2\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j, k \in \mathcal{S}_{\text{rob}}) \vee (j = k \wedge j, k \in d_{\text{irr}} \wedge i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j, k \in \mathcal{S}_{\text{vul}}) \vee (j = k \wedge j, k \in d_{\text{irr}} \wedge i \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha\beta\gamma^2) & (i \in \mathcal{S}_{\text{irr}}) \wedge \{(j \in \mathcal{S}_{\text{rob}} \wedge k \in \mathcal{S}_{\text{vul}}) \vee (j \in \mathcal{S}_{\text{vul}} \wedge k \in \mathcal{S}_{\text{rob}})\} \\ 0 & (j \neq k) \wedge (j \in \mathcal{S}_{\text{irr}} \vee k \in \mathcal{S}_{\text{irr}}) \end{cases}. \tag{A198}
\end{aligned}$$

*Proof.* We first note that

$$\mathbb{E}[x_i^2] = \mathbb{E}[(yx_i)^2] = \mathbb{E}[(yx_i - \mathbb{E}[yx_i])^2] + \mathbb{E}[yx_i]^2 = \begin{cases} \mathcal{O}(\alpha^2) & (i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2) & (i \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^2) & (i \in \mathcal{S}_{\text{irr}}) \end{cases}, \quad (\text{A199})$$

$$\mathbb{E}[yx_i^3] = \mathbb{E}[(yx_i)^3] \quad (\text{A200})$$

$$= \mathbb{E}[(yx_i - \mathbb{E}[yx_i])^3] + 3\mathbb{E}[(yx_i)^2]\mathbb{E}[yx_i] - 2\mathbb{E}[yx_i]^3 \quad (\text{A201})$$

$$= \begin{cases} \mathcal{O}(\alpha^3) & (i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^3) & (i \in \mathcal{S}_{\text{vul}}) \\ 0 & (i \in \mathcal{S}_{\text{irr}}) \end{cases}, \quad (\text{A202})$$

$$\mathbb{E}[x_i^4] = \mathbb{E}[(yx_i - \mathbb{E}[yx_i])^4] + 4\mathbb{E}[yx_i^3]\mathbb{E}[yx_i] - 6\mathbb{E}[x_i^2]\mathbb{E}[yx_i]^2 + 3\mathbb{E}[yx_i]^4 \quad (\text{A203})$$

$$= \begin{cases} \mathcal{O}(\alpha^4) & (i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^4) & (i \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^4) & (i \in \mathcal{S}_{\text{irr}}) \end{cases}. \quad (\text{A204})$$

(a) For  $(i \neq j) \wedge (i \in \mathcal{S}_{\text{irr}} \vee j \in \mathcal{S}_{\text{irr}})$ ,  $\mathbb{E}[x_j x_i] = \mathbb{E}[x_j]\mathbb{E}[x_i] = 0$ . Using the Cauchy-Schwarz inequality,

$$\mathbb{E}[x_j x_i] \leq \sqrt{\mathbb{E}[x_j^2]}\sqrt{\mathbb{E}[x_i^2]} \quad (\text{A205})$$

$$= \begin{cases} \mathcal{O}(\alpha^2) & (i, j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2) & (i, j \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\gamma^2) & (i, j \in \mathcal{S}_{\text{irr}}) \wedge (i = j) \\ \mathcal{O}(\alpha\beta) & (i \in \mathcal{S}_{\text{rob}} \wedge j \in \mathcal{S}_{\text{vul}}) \vee (i \in \mathcal{S}_{\text{vul}} \wedge j \in \mathcal{S}_{\text{rob}}) \end{cases}. \quad (\text{A206})$$

(b) For  $j \in \mathcal{S}_{\text{irr}}, j = i$ ,  $\mathbb{E}[yx_j x_i^2] = \mathbb{E}[y]\mathbb{E}[x_i^3] = 0$ . For  $j \in \mathcal{S}_{\text{irr}}, j \neq i$ ,  $\mathbb{E}[yx_j x_i^2] = \mathbb{E}[x_j]\mathbb{E}[yx_i^2] = 0$ . Using the Cauchy-Schwarz inequality,

$$\mathbb{E}[yx_j x_i^2] \leq \sqrt{\mathbb{E}[x_j^2]}\sqrt{\mathbb{E}[x_i^4]} = \begin{cases} \mathcal{O}(\alpha^3) & (i, j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^3) & (i, j \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha^2\beta) & (i \in \mathcal{S}_{\text{rob}} \wedge j \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha\beta^2) & (i \in \mathcal{S}_{\text{vul}} \wedge j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\alpha\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j \in \mathcal{S}_{\text{vul}}) \end{cases}. \quad (\text{A207})$$

(c) For  $(j \neq k) \wedge (j \in \mathcal{S}_{\text{irr}} \vee k \in \mathcal{S}_{\text{irr}})$ ,  $\mathbb{E}[x_j x_k x_i^2] = 0$ . For  $j = k$ , using the Cauchy-Schwarz inequality,

$$\mathbb{E}[x_j x_k x_i^2] \leq \sqrt{\mathbb{E}[x_j^4]}\sqrt{\mathbb{E}[x_i^4]} = \begin{cases} \mathcal{O}(\gamma^4) & (j = k) \wedge (i, j, k \in \mathcal{S}_{\text{irr}}) \\ \mathcal{O}(\alpha^2\gamma^2) & (j = k) \wedge (j, k \in \mathcal{S}_{\text{irr}} \wedge i \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2\gamma^2) & (j = k) \wedge (j, k \in \mathcal{S}_{\text{irr}} \wedge i \in \mathcal{S}_{\text{vul}}) \end{cases}. \quad (\text{A208})$$

Using the Cauchy-Schwarz inequality,

$$\mathbb{E}[x_j x_k x_i^2] \leq \sqrt{\mathbb{E}[x_j^2]}\sqrt{\mathbb{E}[x_k^2]}\sqrt{\mathbb{E}[x_i^4]} \quad (\text{A209})$$

$$= \begin{cases} \mathcal{O}(\alpha^4) & (i, j, k \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^4) & (i, j, k \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha^3\beta) & (i \in \mathcal{S}_{\text{rob}}) \wedge \{(j \in \mathcal{S}_{\text{rob}} \wedge k \in \mathcal{S}_{\text{vul}}) \vee (j \in \mathcal{S}_{\text{vul}} \wedge k \in \mathcal{S}_{\text{rob}})\} \\ \mathcal{O}(\alpha\beta^3) & (i \in \mathcal{S}_{\text{vul}}) \wedge \{(j \in \mathcal{S}_{\text{rob}} \wedge k \in \mathcal{S}_{\text{vul}}) \vee (j \in \mathcal{S}_{\text{vul}} \wedge k \in \mathcal{S}_{\text{rob}})\} \\ \mathcal{O}(\alpha^2\beta^2) & (i \in \mathcal{S}_{\text{rob}} \wedge j, k \in \mathcal{S}_{\text{vul}}) \vee (i \in \mathcal{S}_{\text{vul}} \wedge j, k \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\alpha^2\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j, k \in \mathcal{S}_{\text{rob}}) \\ \mathcal{O}(\beta^2\gamma^2) & (i \in \mathcal{S}_{\text{irr}} \wedge j, k \in \mathcal{S}_{\text{vul}}) \\ \mathcal{O}(\alpha\beta\gamma^2) & (i \in \mathcal{S}_{\text{irr}}) \wedge \{(j \in \mathcal{S}_{\text{rob}} \wedge k \in \mathcal{S}_{\text{vul}}) \vee (j \in \mathcal{S}_{\text{vul}} \wedge k \in \mathcal{S}_{\text{rob}})\} \end{cases}. \quad (\text{A210})$$

□

G PROOF OF **THEOREM 3.7** (TRADE-OFF)

**Theorem 3.7** (Accuracy–robustness trade-off). Assume  $d_{\text{rob}} = 1$ ,  $d_{\text{vul}} = d - 1$ , and  $d_{\text{irr}} = 0$ . In addition to **Assumption 3.2**, for  $(\mathbf{x}, y) \sim \mathcal{D}^{\text{te}}$ , suppose that  $yx_i$  takes  $\alpha$  with probability  $p > 0.5$  and  $-\alpha$  with probability  $1 - p$  for  $i \in \mathcal{S}_{\text{rob}}$ . Moreover,  $yx_i$  takes  $\beta$  with probability one for  $i \in \mathcal{S}_{\text{vul}}$ . Define  $\tilde{f}(\mathbf{P}, \mathbf{Q}) := \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}^{\text{te}}} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}, \mathbf{Q})]_{d+1, N+1}]$ . Then, there exist strictly positive functions  $g_1(d, \alpha, \beta)$  and  $g_2(d, \alpha, \beta)$  such that

$$\tilde{f}(\mathbf{P}^{\text{std}}, \mathbf{Q}^{\text{std}}) = \begin{cases} g_1(d, \alpha, \beta)(\alpha + (d-1)\beta) & (\text{w.p. } p) \\ g_1(d, \alpha, \beta)(-\alpha + (d-1)\beta) & (\text{w.p. } 1-p) \end{cases}, \quad (\text{A210})$$

$$\tilde{f}(\mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}}) \leq g_2(d, \alpha, \beta)\{-(2p-1)\alpha^2 + (d-1)\beta^2\} \quad (\text{w.p. } 1-p). \quad (\text{A211})$$

*Proof.* Using  $\mathbf{b}$  and  $\mathbf{A}$  defined in **Appendix E**, we can rearrange  $\tilde{f}(\mathbf{P}, \mathbf{Q})$  as

$$\tilde{f}(\mathbf{P}, \mathbf{Q}) := \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}, \mathbf{Q})]_{d+1, N+1}] \quad (\text{A212})$$

$$= \frac{1}{N} \mathbf{b}^\top \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top] \mathbf{A} y_{N+1} \mathbf{x}_{N+1}. \quad (\text{A213})$$

**Standard Transformer.** Similarly to the proof of **Theorem 3.5**, using some positive function  $g(d, \alpha, \beta) > 0$ , we can represent  $\mathbb{E}[\mathbf{b}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{A} / N] = [g(d, \alpha, \beta) \cdots g(d, \alpha, \beta)]$ . Thus,

$$\frac{1}{N} \mathbf{b} \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top] \mathbf{A} y_{N+1} \mathbf{x}_{N+1} = \begin{bmatrix} g(d, \alpha, \beta) \\ \vdots \\ g(d, \alpha, \beta) \end{bmatrix}^\top y_{N+1} \mathbf{x}_{N+1} \quad (\text{A214})$$

$$= g(d, \alpha, \beta) y_{N+1} \sum_{i=1}^d x_{N+1, i} \quad (\text{A215})$$

$$= \begin{cases} \alpha + (d-1)\beta & (\text{w.p. } p) \\ -\alpha + (d-1)\beta & (\text{w.p. } 1-p) \end{cases}. \quad (\text{A216})$$

**Adversarially Trained Transformer.** Now,

$$\begin{aligned} & \frac{1}{N} \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top] \\ &= \begin{bmatrix} \mathbb{E}[(y\mathbf{x})(y\mathbf{x}^\top)] & \mathbb{E}[y\mathbf{x}] \\ \mathbb{E}[y\mathbf{x}^\top] & 1 \end{bmatrix} \end{aligned} \quad (\text{A217})$$

$$= \begin{bmatrix} \alpha^2 & (2p-1)\alpha\beta & \cdots & (2p-1)\alpha\beta & (2p-1)\alpha \\ (2p-1)\alpha\beta & \beta^2 & \cdots & \beta^2 & \beta \\ (2p-1)\alpha\beta & \beta^2 & \cdots & \beta^2 & \beta \\ \vdots & & & & \\ (2p-1)\alpha\beta & \beta^2 & \cdots & \beta^2 & \beta \\ (2p-1)\alpha & \beta & \cdots & \beta & 1 \end{bmatrix}. \quad (\text{A218})$$

Thus,

$$\frac{1}{N} \mathbf{b}^\top \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top] \mathbf{A} = \begin{bmatrix} \alpha\{\alpha + (d-1)(2p-1)\beta + (2p-1)\} \\ \beta\{(2p-1)\alpha + (d-1)\beta + 1\} \\ \vdots \\ \beta\{(2p-1)\alpha + (d-1)\beta + 1\} \end{bmatrix}^\top. \quad (\text{A219})$$

Therefore,

$$\begin{aligned} & \frac{1}{N} \mathbf{b}^\top \mathbb{E}_{\{(\mathbf{x}_n, y_n)\}_{n=1}^N} [\mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top] \mathbf{A} y_{N+1} \mathbf{x}_{N+1} \\ &= \begin{bmatrix} \alpha\{\alpha + (d-1)(2p-1)\beta + (2p-1)\} \\ \beta\{(2p-1)\alpha + (d-1)\beta + 1\} \\ \vdots \\ \beta\{(2p-1)\alpha + (d-1)\beta + 1\} \end{bmatrix}^\top \begin{bmatrix} y_{N+1} x_{N+1,1} \\ \beta \\ \vdots \\ \beta \end{bmatrix} \end{aligned} \quad (\text{A219})$$

$$= \begin{cases} \alpha^2\{\alpha + (d-1)(2p-1)\beta + (2p-1)\} \\ \quad + (d-1)\beta^2\{(2p-1)\alpha + (d-1)\beta + 1\} & (\text{w.p. } p) \\ -\alpha^2\{\alpha + (d-1)(2p-1)\beta + (2p-1)\} \\ \quad + (d-1)\beta^2\{(2p-1)\alpha + (d-1)\beta + 1\} & (\text{w.p. } 1-p) \end{cases}. \quad (\text{A220})$$

In particular,

$$\begin{aligned} & -\alpha^2\{\alpha + (d-1)(2p-1)\beta + (2p-1)\} + (d-1)\beta^2\{(2p-1)\alpha + (d-1)\beta + 1\} \\ &= \{(2p-1)\alpha + (d-1)\beta + 1\}(-C\alpha^2 + (d-1)\beta^2), \end{aligned} \quad (\text{A221})$$

where

$$C = \frac{\alpha + (d-1)(2p-1)\beta + (2p-1)}{(2p-1)\alpha + (d-1)\beta + 1} > \frac{(2p-1)^2\alpha + (d-1)(2p-1)\beta + (2p-1)}{(2p-1)\alpha + (d-1)\beta + 1} \quad (\text{A222})$$

$$= 2p - 1. \quad (\text{A223})$$

□

## H PROOF OF THEOREM H.1 (NEED FOR LARGER SAMPLE SIZE)

**Theorem H.1** (Need for Larger Sample Size). Assume the same assumptions in [Theorem 3.7](#). Then,

$$\mathbb{E}_{\mathbf{x}_{N+1}, y_{N+1}} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}^{\text{std}}, \mathbf{Q}^{\text{std}})]_{d+1, N+1}] > 0 \quad (\text{w.p. at least } 1 - e^{-pN}). \quad (\text{A224})$$

In addition, suppose that there exists a constant  $0 < C < 1$  such that  $(d-1)\beta + 1 < C\alpha$ . Moreover, assume that  $N$  is an even number. Then, as  $p \rightarrow \frac{1}{2}$  with  $p > \frac{1}{2}$ , for  $4 \leq N \leq \frac{2}{C}$ ,

$$\begin{aligned} & \mathbb{E}_{\mathbf{x}_{N+1}, y_{N+1}} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}})]_{d+1, N+1}] > 0 \\ & \left( \text{w.p. at most } 1 - \frac{0.483}{\sqrt{N}} < 1 - e^{-pN} \right). \end{aligned} \quad (\text{A225})$$

*Proof.* Using  $\mathbf{b}$  and  $\mathbf{A}$  defined in [Appendix E](#), we can calculate

$$\mathbb{E}_{\mathbf{x}_{N+1}, y_{N+1}} [y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}, \mathbf{Q})]_{d+1, N+1}] = \frac{1}{N} \mathbf{b}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{A} \mathbb{E}[y_{N+1} \mathbf{x}_{N+1}]. \quad (\text{A226})$$

Now,

$$\begin{aligned} & \frac{1}{N} \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \\ &= \begin{bmatrix} \alpha^2 & \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} & \cdots & \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} & \frac{1}{N} \sum_{n=1}^N y_n x_{n,1} \\ \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} & \beta^2 & \cdots & \beta^2 & \beta \\ \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} & \beta^2 & \cdots & \beta^2 & \beta \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} & \beta^2 & \cdots & \beta^2 & \beta \\ \frac{1}{N} \sum_{n=1}^N y_n x_{n,1} & \beta & \cdots & \beta & 1 \end{bmatrix}. \end{aligned} \quad (\text{A227})$$

**Standard Transformer.** From the configuration of  $\mathbf{b}$  and  $\mathbf{A}$ , all the entries of  $\mathbf{b}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{A}$  are the same. Since all the entries of  $\mathbb{E}[y_{N+1} \mathbf{x}_{N+1}]$  are positive, with some positive function  $g(d, \alpha, \beta) > 0$ ,

$$\frac{1}{N} \mathbf{b}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{A} \mathbb{E}[y_{N+1} \mathbf{x}_{N+1}] = g(d, \alpha, \beta) \frac{1}{N} \mathbf{1}_{d+1}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{1}_{d+1}. \quad (\text{A228})$$

Now,

$$\begin{aligned} & \frac{1}{N} \mathbf{1}_{d+1}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{1}_{d+1} \\ &= (d-1)^2 \beta^2 + 2(d-1)\beta + 1 + \alpha^2 + \frac{2}{N} \sum_{n=1}^N y_n x_{n,1} + 2(d-1) \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} \end{aligned} \quad (\text{A229})$$

$$= \{(d-1)\beta + 1\}^2 + \alpha^2 + \frac{2\{(d-1)\beta + 1\}}{N} \sum_{n=1}^N y_n x_{n,1} \quad (\text{A230})$$

$$= [\{(d-1)\beta + 1\} - \alpha]^2 + \frac{2\{(d-1)\beta + 1\}}{N} \sum_{n=1}^N (\alpha + y_n x_{n,1}) \quad (\text{A231})$$

$$> 0 \quad (\text{w.p. at least } 1 - (1-p)^N > 1 - e^{-pN}). \quad (\text{A232})$$

**Adversarially Trained Transformer.** Note that  $\mathbb{E}[y_{N+1} \mathbf{x}_{N+1}] = [(2p-1)\alpha \ \beta \ \cdots \ \beta]$ . Thus,

$$\begin{aligned} & \frac{1}{N} \mathbf{1}_{d+1}^\top \mathbf{Z}_0 \mathbf{M} \mathbf{Z}_0^\top \mathbf{I}_d \mathbb{E}[y_{N+1} \mathbf{x}_{N+1}] \\ &= (2p-1)\alpha \left( \alpha^2 + (d-1) \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} + \frac{1}{N} \sum_{n=1}^N y_n x_{n,1} \right) \\ & \quad + (d-1)\beta \left( \frac{\beta}{N} \sum_{n=1}^N y_n x_{n,1} + (d-1)\beta^2 + \beta \right) \end{aligned} \quad (\text{A233})$$

$$\begin{aligned} &= [(2p-1)\alpha^3 + (d-1)\beta^2\{(d-1)\beta + 1\}] \\ & \quad + [(2p-1)\alpha\{(d-1)\beta + 1\} + (d-1)\beta^2] \frac{1}{N} \sum_{n=1}^N y_n x_{n,1}. \end{aligned} \quad (\text{A234})$$

This indicates  $\mathbb{E}_{\mathbf{x}_{N+1}, y_{N+1}}[y_{N+1} [f(\mathbf{Z}_0; \mathbf{P}^{\text{adv}}, \mathbf{Q}^{\text{adv}})]_{d+1, N+1}] > 0$  only if

$$\frac{1}{N} \sum_{n=1}^N y_n x_{n,1} > - \frac{(2p-1)\alpha^3 + (d-1)\beta^2\{(d-1)\beta + 1\}}{(2p-1)\alpha\{(d-1)\beta + 1\} + (d-1)\beta^2}. \quad (\text{A235})$$

Representing  $y_n x_{n,1} = \alpha(2X_n - 1)$  with  $X_n$  taking 1 with probability  $p$  and 0 with probability  $1-p$ ,

$$\begin{aligned} & \frac{1}{N} \sum_{n=1}^N \alpha(2X_n - 1) > - \frac{(2p-1)\alpha^3 + (d-1)\beta^2\{(d-1)\beta + 1\}}{(2p-1)\alpha\{(d-1)\beta + 1\} + (d-1)\beta^2} \\ & \iff \sum_{n=1}^N X_n > \frac{N}{2} \left( 1 - \frac{1}{\alpha} \frac{(2p-1)\alpha^3 + (d-1)\beta^2\{(d-1)\beta + 1\}}{(2p-1)\alpha\{(d-1)\beta + 1\} + (d-1)\beta^2} \right). \end{aligned} \quad (\text{A236})$$

Let  $Y \sim B(N, p)$ , where  $B(N, p)$  is the Binomial distribution. Consider the following probability:

$$\mathbb{P}_{Y \sim B(N, p)} \left[ Y > \frac{N}{2} \left( 1 - \frac{1}{\alpha} \frac{(2p-1)\alpha^3 + (d-1)\beta^2\{(d-1)\beta + 1\}}{(2p-1)\alpha\{(d-1)\beta + 1\} + (d-1)\beta^2} \right) \right]. \quad (\text{A237})$$

When  $p \rightarrow 1/2$ ,

$$\begin{aligned} & \mathbb{P}_{Y \sim B(N, p)} \left[ Y > \frac{N}{2} \left( 1 - \frac{1}{\alpha} \frac{(2p-1)\alpha^3 + (d-1)\beta^2\{(d-1)\beta + 1\}}{(2p-1)\alpha\{(d-1)\beta + 1\} + (d-1)\beta^2} \right) \right] \\ & \rightarrow \mathbb{P}_{Y \sim B(N, 1/2)} \left[ Y > \frac{N}{2} \left( 1 - \frac{(d-1)\beta + 1}{\alpha} \right) \right] \end{aligned} \quad (\text{A238})$$

$$\leq \mathbb{P}_{Y \sim B(N, 1/2)} \left[ Y > \frac{N}{2} (1 - C) \right] \quad (\text{A239})$$

$$\leq \mathbb{P}_{Y \sim B(N, 1/2)} \left[ Y > \frac{N}{2} - 1 \right]. \quad (\text{A240})$$

From Ash (1990), for an integer  $0 < k < N/2$ ,

$$\mathbb{P}_{Y \sim B(N, 1/2)} [Y \leq k] \geq \frac{1}{\sqrt{8N \frac{k}{N} (1 - \frac{k}{N})}} \exp\left(-ND \left(\frac{k}{N} \parallel \frac{1}{2}\right)\right), \quad (\text{A241})$$

where  $D$  is the Kullback–Leibler divergence. Substituting  $k = \frac{N}{2} - 1$ ,

$$\begin{aligned} & \mathbb{P}_{Y \sim B(N, 1/2)} \left[ Y \leq \frac{N}{2} - 1 \right] \\ & \geq \frac{1}{\sqrt{8N \left(\frac{1}{2} - \frac{1}{N}\right) \left\{1 - \left(\frac{1}{2} - \frac{1}{N}\right)\right\}}} \exp\left(-ND \left(\frac{1}{2} - \frac{1}{N} \parallel \frac{1}{2}\right)\right) \end{aligned} \quad (\text{A242})$$

$$= \frac{1}{\sqrt{2\left(1 - \frac{4}{N^2}\right)}} \frac{1}{\sqrt{N}} \exp\left(-ND \left(\frac{1}{2} - \frac{1}{N} \parallel \frac{1}{2}\right)\right). \quad (\text{A243})$$

Note that

$$D\left(\frac{1}{2} - \frac{1}{N} \parallel \frac{1}{2}\right) = \frac{1}{2} \left\{ \left(1 - \frac{2}{N}\right) \ln\left(1 - \frac{2}{N}\right) + \left(1 + \frac{2}{N}\right) \ln\left(1 + \frac{2}{N}\right) \right\}. \quad (\text{A244})$$

For  $N \geq 4$ ,

$$\frac{1}{\sqrt{2\left(1 - \frac{4}{N^2}\right)}} \exp\left(-ND \left(\frac{1}{2} - \frac{1}{N} \parallel \frac{1}{2}\right)\right) > 0.483. \quad (\text{A245})$$

In summary,

$$\mathbb{P}_{Y \sim B(N, 1/2)} \left[ Y > \frac{N}{2} - 1 \right] = 1 - \mathbb{P}_{Y \sim B(N, 1/2)} \left[ Y \leq \frac{N}{2} - 1 \right] \leq 1 - \frac{0.483}{\sqrt{N}}. \quad (\text{A246})$$

□