Ensuring Calibration Robustness in Split Conformal Prediction Under Adversarial Attacks

Xunlei Qian [*1] **Yue Xing** [*1]

Abstract

Conformal prediction (CP) provides distribution-free, finite-sample coverage guarantees but critically relies on exchangeability, a condition often violated under distribution shift. We study the robustness of split conformal prediction under adversarial perturbations at test time, focusing on both coverage validity and the resulting prediction efficiency. Our theoretical analysis characterizes how the strength of adversarial perturbations during calibration affects the coverage gap relative to the nominal coverage level under adversarial test conditions. We further examine the impact of adversarial training at the model-training stage. Experiments support our theory: (i) Prediction coverage varies monotonically with the calibration-time attack strength, enabling the use of nonzero calibration-time attack to predictably control coverage under adversarial tests; (ii) the marginal coverage can remain within a user-specified tolerance band around the nominal coverage level and (iii) adversarial training at the training stage produces tighter prediction sets that improve efficiency while maintaining coverage validity.

1. Introduction

With the rapid adoption of deep learning in high-stakes domains, such as survival analysis (Candes et al., 2023) and chest X-ray report generation (Gui et al., 2024), reliable uncertainty quantification has become essential for safe deployment in high-stakes domains. Conformal prediction (CP) (Vovk et al., 2005) offers a powerful, distribution-free framework that provides finite-sample validity by constructing prediction sets calibrated to a nominal coverage level. For example, CP can support clinical decision workflows in which physicians must deploy complex models for diagnosis (Banerji et al., 2023; Kiyani et al., 2025; Hullman et al., 2025).

Despite its flexibility, CP typically relies on independent and identically distributed (i.i.d.) or, more generally, at least exchangeable data. These assumptions are often violated in practice (Barber et al., 2023). When exchangeability is broken, the conformity-score distribution in the calibration set need not match that of the test data due to distribution shift commonly observed in real-world settings (Moreno-Torres et al., 2012; Sugiyama et al., 2007; Koh et al., 2021). A large body of work proposes mitigation strategies within the classical CP paradigm: some methods address random perturbations at test time (Gendler et al., 2022; Yan et al., 2024), while others demonstrate that adversarial perturbations can be particularly harmful for deep models (Kos and Song, 2017). These observations motivate the study of CP under non-exchangeable conditions.

Among such shifts, adversarial attacks are a prominent form of distribution shift in which inputs are deliberately perturbed in worst-case directions to degrade performance (Biggio et al., 2013; Goodfellow et al., 2015). Although such perturbations are often imperceptible to humans (Wang et al., 2019), the deep neural network or even transformer-based models like Vision Transformers (ViTs) can suffer substantial drops in performance compared with unperturbed inputs (Shao et al., 2021; Aldahdooh et al., 2021; Narodytska and Kasiviswanathan, 2017; Miller et al., 2020). Prior work spans attack generation (Xiao et al., 2018; Assion et al., 2019), adversarial training (Shafahi et al., 2019; Andriushchenko and Flammarion, 2020; Shafahi et al., 2020), and adversarial robustness (Bai et al., 2021; Silva and Najafirad, 2020).

In this paper, we examine test-time adversarial perturbations through the lens of conformal prediction. We analyze how such attacks affect marginal coverage and efficiency, how calibration under adversarial perturbations can improve robustness at the test time, and how adversarial training improves the efficiency of split conformal prediction. Our contributions are as follows:

- We provide a theoretical analysis of how the strength

[1]Department of Statistics and Probability, Michigan State University, East Lansing, USA. Correspondence to: Xunlei Qian <qianxunl@msu.edu>, Yue Xing <yuexing1@msu.edu>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea.

of adversarial perturbations applied to the calibration set affects marginal coverage on both clean and adversarially perturbed test data. In particular, we show that marginal coverage under a fixed test-time attack level is a monotone (non-decreasing) function of the calibration attack strength. (Theorem 3.2).

- We show that a proper adversarial perturbation during calibration yields more robust predictions under test-time attacks: for a 90% nominal level, CP attains reasonable coverage (e.g., $87\% - 93\%$) across a wider range of test-time attack strengths (Theorem 3.4).
- We show that adversarial training can reduce prediction set size for Split conformal prediction, thereby improving both robustness and efficiency for deep neural networks (Theorem 3.5).
- Experiments are conducted to verify the observations from the above theorems.

2. Preliminaries

The following section provides the general framework of split conformal prediction (Split CP) which we will adopt in our theorem and experiments.

2.1. Split Conformal Prediction

Split CP provides distribution-free, finite-sample coverage guarantees at user-specified levels for both classification and regression (Lei et al., 2018; Romano et al., 2020). We adopt split CP as our primary procedure for constructing prediction sets.

Consider a multi-class classification problem, where we aim to train a model $f : X \to T$, with $T = \{1, 2, \ldots, K\}$ representing the set of class labels and X denoting the input space. To construct prediction sets with coverage guarantees, we adopt the *Split Conformal Prediction* framework (Romano et al., 2020; Lei et al., 2018). The data is partitioned into three disjoint subsets: a training set I_1 with sample size n_1, a calibration set I_2 with sample size n_2, and a test set I_3 with sample size n_3.

We define the nonconformity score as

$$S(x, y) = 1 - f_y(x),$$

where $f_y(x) = \mathbb{P}(y \mid x)$ is the estimated probability assigned by the trained model to class y given x.

Given a new testing input x_{test} from testing set I_3 , the prediction set is constructed as

$$\begin{aligned} C(x_{test}) &= \{y \in \{1, 2, \ldots, K\} : f_y(x_{test}) \geq 1 - Q\} \\ &= \{y \in \{1, 2, \ldots, K\} : 1 - f_y(x_{test}) \leq Q\}, \end{aligned}$$

where Q is the $(1 - \alpha)\,(1 + 1/|I_2|)$-quantile of the nonconformity scores $\{S(x_i, y_i) : i \in I_2\}$.

Remark 2.1. There are several choices for the nonconformity score, including the HPS score proposed in (Lei et al., 2018) and the APS score introduced in (Romano et al., 2020):

$$\begin{aligned} S_{\mathrm{HPS}}(x, y) &= 1 - f_y(x), \\ S_{\mathrm{APS}}(x, y) &= \sum_{y' \in [K]} \hat{\pi}_{y'}(x) \cdot \mathbf{1}_{\{f_{y'}(x) > f_y(x)\}} + f_y(x) \cdot u, \end{aligned}$$

where $u \sim \mathcal{U}(0, 1)$ is drawn from the uniform distribution on the unit interval. We focus on the HPS nonconformity score because it allows a more direct analysis of how adversarial attacks affect prediction top-1 accuracy.

Algorithm 1 A General Form of Adversarial Training

Input: data $\{(x_i, y_i)\}_{i=1}^n$, attack bound ϵ, training steps T, initialization $\theta^{(0)}$, step size η
Initialize: $t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  for $i = 1$ **to** n **do**
    Compute adversarial perturbation:
      $\delta_i \leftarrow A_\epsilon(f_{\theta^{(t-1)}}, x_i, y_i)$
    Form adversarial example:
      $\tilde{x}_i \leftarrow x_i + \delta_i$
  end for
  Compute the adversarial empirical risk:
    $L^{(t)} \leftarrow \frac{1}{n} \sum_{i=1}^n \ell\big(f_{\theta^{(t-1)}}(\tilde{x}_i), y_i\big)$
  Update model parameters (gradient descent):
    $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta \nabla_\theta L^{(t)}$
until $t = T$
Output: $\theta^{(T)}$

To evaluate Split CP, we report validity and efficiency. Validity refers to the proportion of test instances whose true label is contained in the conformal prediction set and should be near the nominal coverage level $1 - \alpha$; we summarize it via empirical coverage on the test set. Efficiency is quantified by the average prediction set size.

Since Split CP is a post-training procedure, robustness to test-time attacks must come from the base model itself. We therefore adopt adversarial training, introduced below, before presenting our main theoretical results:

2.2. Adversarial training

Adversarial training is formulated as follows (Xing et al., 2021): let ℓ denote the loss function and let $f_\theta : \mathbb{R}^d \to \mathbb{R}^K$ be the model with parameters θ. The (population) adversarial loss is defined as $R_f(\theta, \epsilon) := \mathbb{E}\left[l\left(f_\theta\left[x + A_\epsilon(f_\theta, x, y)\right], y\right)\right]$, where A_ϵ is an attack of strength $\epsilon > 0$ and is designed to worsen the loss in the

following way:

$$A_\epsilon(f_\theta, x, y) := \arg \max_{z \in \mathcal{R}(0,\epsilon)} \{l(f_\theta(x+z), y)\}.$$

In the above, z is constrained to the perturbation set $\mathcal{R}(0, \epsilon)$; throughout this paper we focus on ℓ_∞ attacks, i.e., the ℓ_∞-ball of radius ϵ centered at the origin. We use the ℓ_∞ attack for both adversarial training, calibration-time and test-time attacks in experiments, and our proofs are carried out under ℓ_2 attack assumption. We follow the general adversarial-training framework of (Xing et al., 2021):

Briefly speaking, in Algorithm 1, in each iteration, we first calculate the attack of each sample given the current model, and then update the model's parameters based on the loss value calculated from the attacked samples.

We use ℓ_∞-FGSM for both test-time attacks and adversarial training because it matches our ℓ_∞ experimental threat model and requires only a single gradient step, enabling scalable, reproducible runs across many ϵ values and datasets. Its simplicity keeps training and evaluation aligned, so observed effects are attributable to ϵ_{cal} rather than attack optimization.

2.3. The Whole Pipeline

After introducing Split CP and adversarial training, we present the whole pipeline as follows:

- Training Stage: Due to potential corruptions in the testing stage, we use adversarial training with attack strength ϵ_{train} on the training dataset I_1 with size n_1 to train the model.
- Calibration Stage: We inject an adversarial attack in each sample with attack strength ϵ_{cal} for the calibration data I_2 and construct the Split CP to achieve the nominal coverage level $1-\alpha$.
- Testing Stage: Apply Split CP on I_3, where samples may carry attacks of unknown strength ϵ_{test}.

Our target in this paper is to analyze the impact of adversarial training and adversarial attack towards the Split CP procedure. In particular, we study whether the calibration set I_2 contains adversarial attack or not, and whether we use clean training or adversarial training in the training stage.

3. Main Results

In this section, we present our main theoretical results on the impact of adversarial attacks to the calibration set within Split CP. We analyze how such attacks influence the validity and efficiency of prediction sets when prediction sets are evaluated on adversarially perturbed test inputs in Theorem 3.2 and Theorem 3.4. Furthermore, we provide a formal proposition that theoretically justify the observed reduction in prediction set size under adversarial training in training stage in Theorem 3.5.

3.1. Assumptions

We collect all assumptions used in the theoretical results in one place. Each theorem statement indicates the minimal subset of assumptions it requires; every theorem in Section 3 is derived under at most (A1)–(A5) below.

(A1) Data regularity. The labeled sample $\{(x_i, y_i)\}_{i=1}^n \cup \{(x_{\text{test}}, y_{\text{test}})\}$ is exchangeable under P on $\mathcal{X} \times \mathcal{Y}$. The trained classifier $\widehat{f} : \mathcal{X} \to \Delta^{K-1}$ and is independent of the calibration fold I_2 and the test point.

(A2) Score smoothness. The nonconformity score $S(x, y) := 1 - \widehat{f}_y(x)$ is differentiable in x on the support of P, and there exists $L < \infty$ such that $\|\nabla_x S(x, y)\| \le L$ for P-almost every (x, y).

(A3) Small-perturbation regime. The calibration and test attack budgets satisfy $\max(\varepsilon_{\text{cal}}, \varepsilon_{\text{test}}) = o(1)$. Under (A2), first-order Taylor expansions of S around clean inputs and of the finite-sample quantile $Q_S(\varepsilon_{\text{cal}})$ around $\varepsilon_{\text{cal}} = 0$ are valid up to $O(\varepsilon^2)$, where $\varepsilon := \max(\varepsilon_{\text{cal}}, \varepsilon_{\text{test}})$.

(A4) Non-degenerate calibration quantile. Define $\beta = (1-\alpha)(1 + 1/|I_2|)$. The finite-sample $(1-\alpha)$th-quantile $Q := Q_\beta(\{S(x_i, y_i) : i \in I_2\})$ satisfies $1 - Q > 0$ P-almost surely.

(A5) Score Stochastic Dominance. Let $S^\star(x, y; \epsilon) := 1 - \widehat{f}^\star_y(x + \delta), \star \in \{\text{adv}, \text{clean}\}$, denote the nonconformity score of classifier $\star$ evaluated on a point perturbed by $\|\delta\|_\infty \le \epsilon$. We assume the adversarially trained score distribution first-order stochastically smaller than the cleanly trained one: $F_{S^{\text{adv}}}(t) \ge F_{S^{\text{clean}}}(t)$, for all $t \in [0, 1]$, where $F_{S^\star}$ denotes the CDF of $S^\star$ under $(x, y) \sim P$, δ drawn adversarially.

In the above, (A1) is the standard Split CP exchangeability setting widely used in existing literature (Vovk et al., 2005; Lei et al., 2018). The smoothness condition (A2) is a common assumption in deep-learning literature, e.g., (Bengio et al., 2017). In (A3), with the small perturbation assumption $\max(\epsilon_{\text{cal}}, \epsilon_{\text{test}}) = o(1)$, one can use the first-order Taylor expansions of S to extract the attack and separate it from the clean S. It is a common setting in adversarial training literature, e.g., the widely-used 8/255 ℓ_∞ attack in image data (Rice et al., 2020). Assumption (A4) is a mild non-degeneracy condition on the calibration quantile, and it is always valid as long as the distribution of S does not

cluster on 0 or 1. It is needed in Theorem 3.5 so that the $(1 - P_{y,\text{true}})/(1 - Q)$ term in the prediction-set-size bound is finite. Assumption (A5) assumes that the adversarially-trained nonconformity score distribution is stochastically smaller than the cleanly-trained one (Grabinski et al., 2022; Javanmard and Mehrabi, 2024). It is used in Theorem 3.5 to derive an inequality that yields a strictly smaller expected Split CP prediction-set size under adversarial training than under clean training.

3.2. Validity of Split CP at the post-training stage

In the section, we first present a previous result from (Oliveira et al., 2024). The following Proposition 3.1 provides an error bound on the coverage gap between the validity of the split CP and the accuracy of the nominal coverage level $1 - \alpha$. Unlike common literature in which only one side of the error bound is presented, (Oliveira et al., 2024) provides both the upper bound and the lower bound. This is essential to help us further construct the pair of upper and lower bounds for Split CP generated by adversarially attacked calibration set using adversarial testing data.

Proposition 3.1. *Let (x_i, y_i) for $i = 1, \ldots, n + 1$ be exchangeable sample data, and suppose there exist constants $e_{cal}, d_{cal}, e_{train} \in (0, 1)$ for which the following two conditions hold:*

Calibration Concentration.

$$P\left[\left|\frac{1}{n_2}\sum_{(x_i,y_i)\in I_2}\mathbf{1}\{f_{y_i}(x_i) \le Q\} - P_{q,train}\right| \le e_{cal}\right] \ge 1 - d_{cal},$$

where (x_*, y_*) is any data point independent of the training, calibration, and testing data, $P_{q,train} = P[f_{y_*}(x_*) \le q_{train} \mid (x_j, y_j)_{j\in|I_1|}]$, the mapping $q : (X \times Y)^{I_1} \to \mathbb{R}$ is any measurable function, and $q_{train} = q((x_i, y_i), i \in I_1)$.

Test-Time Stability.

$$|\,\mathbb{P}[f_{x_k}(y_k) \le q_{train}] - \mathbb{P}[f_{x_*}(y_*) \le q_{train}]\,| \le e_{train}, \quad \text{for } (x_k, y_k) \in I_3.$$

Then the prediction set $C(\cdot)$ given by Split CP satisfies

$$\left|\mathbb{P}(Y_{n+1} \in C(X_{n+1})) - (1 - \alpha)\right| \le e_{train} + d_{cal} + e_{cal}.$$

Given the two-sided error bound result in Proposition 3.1, Theorem 3.2 below analyzes the gap between the prediction accuracy of Split CP and nominal coverage the target accuracy $1 - \alpha$. The theorem considers the case where there is a test-time attack with strength ϵ_{test} and the calibration attack strength is taken as ϵ_{cal}.

Theorem 3.2 (First-order coverage sensitivity identity)**.** *Assume* (A1)–(A3)*, and let g denote the density of $\widehat{f}_y(x)$ under P, assumed twice differentiable in a neighbourhood of $Q_\alpha(\widehat{f}_{y_{\text{cal}}}(x_{\text{cal}}))$ with $g(Q_\alpha) > 0$. Let $p \in [1, \infty]$ and let q denote its Hölder conjugate, $1/p + 1/q = 1$. Let $A : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ be a fixed measurable attack-direction map with $\|A(x, y)\|_p \le 1$, applied with the same rule to both calibration and test inputs. Define the* direction-specific score-drop coefficient

$$b_A(x, y) := -A(x, y)^\top \nabla_x \widehat{f}_y(x), \tag{1}$$

and set

$$c_A := \mathbb{E}\Big[b_A(x_{\text{test}}, y_{\text{test}}) \,\Big|\, \widehat{f}_{y_{\text{test}}}(x_{\text{test}}) = Q_\alpha(\widehat{f}_{y_{\text{cal}}}(x_{\text{cal}}))\Big]. \tag{2}$$

Let $C_{\epsilon_{\text{cal}}}(x_{\text{test}} + \epsilon_{\text{test}} A_{\text{test}})$ denote the Split CP prediction set obtained from a calibration fold attacked at strength ϵ_{cal} in direction A and evaluated at $x_{\text{test}} + \epsilon_{\text{test}} A_{\text{test}}$, where $A_{\text{test}} = A(x_{\text{test}}, y_{\text{test}})$. Then the signed coverage gap satisfies the exact first-order identity

$$\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(x_{\text{test}} + \epsilon_{\text{test}} A_{\text{test}})) - (1 - \alpha) \tag{3}$$

$$= -(\epsilon_{\text{test}} - \epsilon_{\text{cal}})\, c_A\, g(Q_\alpha) + R + O\big(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2\big), \tag{4}$$

with residual $|R| \le e_{\text{train}} + d_{\text{cal}} + e_{\text{cal}}$ as defined in Proposition 3.1.

The proof of Theorem 3.2 is in Appendix B.1. A brief intuition is as follows. Under mild regularity conditions, specifically, assuming the smoothness of the probability density function of $g(f_y(x))$ and the smoothness of $G(f_y(x))$, the coverage gap between the prediction set generated from an attacked calibration set and the nominal coverage level is an increasing function of ϵ_{cal} when the ϵ_{test} range is held fixed. In particular, with ϵ_{test} fixed, larger ϵ_{cal} induces larger prediction sets and hence smaller top-1 accuracy shortfall. It is easy to see that when $\epsilon_{cal} = \epsilon_{test}$, the problem reduces to the exchangeable scenario treated in Proposition 3.1.

Theorem 3.2 fixes a single attack direction A, whereas standard adversarial attacks are defined as the worst-case attack within an ℓ_p-ball, recovering FGSM at $p = \infty$ and PGD at $p = 2$. Given Theorem 3.2, the next corollary applies the first-order identity to this worst-case direction by maximizing b_A over $\|A\|_p \le 1$, converting the signed direction-specific identity (3) into a two-sided absolute coverage bound that holds uniformly over every admissible A.

Corollary 3.3 (Worst-case bound; FGSM and PGD)**.** *Under the conditions of Theorem 3.2, define the worst-case score-drop coefficient over the ℓ_p-ball,*

$$b^\star(x, y) := \sup_{\|A\|_p \le 1} b_A(x, y) = \|\nabla_x \widehat{f}_y(x)\|_q, \tag{5}$$

and the corresponding conditional expectation $c^\star := \mathbb{E}[b^\star(x_{\text{test}}, y_{\text{test}}) \mid \widehat{f}_{y_{\text{test}}}(x_{\text{test}}) = Q_\alpha]$. Then for any at-

tack direction A with $\|A\|_p \le 1$,

$$\left|\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(x_{\text{test}} + \epsilon_{\text{test}} A_{\text{test}})) - (1-\alpha)\right| \quad (6)$$

$$\le\ |\epsilon_{\text{test}} - \epsilon_{\text{cal}}|\, c^\star\, g(Q_\alpha)\ +\ E\ +\ O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2), \quad (7)$$

where $E = e_{\text{train}} + d_{\text{cal}} + e_{\text{cal}}$ is from Proposition 3.1. Two standard instantiations are:

- ℓ_2 ***attack*** **(PGD)***: $p = 2$, $q = 2$, $b^\star(x,y) = \|\nabla_x \widehat{f}_y(x)\|_2$, achieved by $A^\star(x,y) = -\nabla_x \widehat{f}_y(x)/\|\nabla_x \widehat{f}_y(x)\|_2$.*
- ℓ_∞ ***attack*** **(FGSM)***: $p = \infty$, $q = 1$, $b^\star(x,y) = \|\nabla_x \widehat{f}_y(x)\|_1$, achieved by $A^\star(x,y) = -\operatorname{sign}(\nabla_x \widehat{f}_y(x))$.*

In both cases $b^\star(x,y) \ge 0$ everywhere and $c^\star > 0$ whenever $\widehat{f}$ is non-constant on the support of P.The bound (6) holds for all signs of $\epsilon_{\text{test}} - \epsilon_{\text{cal}}$.

Besides Theorem 3.2 and Proposition 3.1, we also study how to attain a tolerance band around the target nominal coverage when ϵ_{test} is unknown. Since exact attainment of $1-\alpha$ is impossible without knowing ϵ_{test}, we consider a practical band (e.g., $87\% - 93\%$ for a 90% target). Theorem 3.4 below connects ϵ_{cal}, the tolerance width, the target level, and ϵ_{test}, yielding conditions under which coverage falls within the desired band over a range of test-time attacks. This provides a principled rule for selecting the smallest ϵ_{cal} that achieves the tolerance across the anticipated ϵ_{test} range.

Theorem 3.4. *Assume (A1)–(A3), with $c_A\, g(Q_\alpha(f_{y_{cal}}(x_{cal}))) > 0$, and let $E := e_{train} + d_{cal} + e_{cal}$ denote the non-adversarial Split CP residual bound from Proposition 3.1. Fix a tolerance $\tau > E$. Under the setup of Theorem 3.2, for every calibration-attack strength ϵ_{cal} and every test-time attack strength ϵ_{test} satisfying*

$$\epsilon_{test}\ \in\ \epsilon_{cal} + \left[-\frac{\tau - E}{c_A\, g(Q_\alpha(f_{y_{cal}}(x_{cal})))},\ \frac{\tau - E}{c_A\, g(Q_\alpha(f_{y_{cal}}(x_{cal})))} \right],$$

the Split CP coverage probability satisfies

$$|\mathbb{P}(y_{test} \in C_{\epsilon_{cal}}(x_{test} + \epsilon_{test} A_{test})) - (1-\alpha)| \le \tau + O\big(\epsilon_{cal}^2 + \epsilon_{test}^2\big)\,.$$

The admissible test-attack region is a symmetric interval about ϵ_{cal} with half-width $(\tau - E)/\big[c_A\, g(Q_\alpha(f_{y_{cal}}))\big]$ and total length $2(\tau - E)/\big[c_A\, g(Q_\alpha(f_{y_{cal}}))\big]$.

After explaining the impact of adversarial attack in Split CP, we further present the analysis of how adversarial training in the training stage affects the final Split CP result in the following section.

3.3. Impact of adversarial training at the training stage

To investigate the impact of adversarial training on the prediction set size given by the Split CP, we first recall a relevant result from the literature. The following Example 1, summarizing Theorem 3.2 and Theorem 3.4 in (Li and Li, 2024) establishes that, in the presence of test-time adversarial perturbations, adversarially trained models are more robust than those trained on clean data. This guarantee motivates our subsequent analysis of how such robustness translates into improved prediction-set efficiency under Split CP.

Example 1. *For sufficiently large d, consider training a two-layer neural network of the form*

$$F(X) = \big(F_1(X), F_2(X), \ldots, F_k(X)\big),$$
$$F_i(X) := \sum_{r\in[m]} \sum_{p\in[P]} \widetilde{\mathrm{ReLU}}\big(\langle w_{i,r}, x_p\rangle\big),$$

where $X = (x_p)_{p\in[P]} \in (\mathbb{R}^d)^P$, $w_{i,r} \in \mathbb{R}^d$ are trainable weights, and $m = \mathrm{polylog}(d)$ denotes the network width. Suppose the model is initialized randomly and trained for $T = \Theta(\mathrm{poly}(d)/\eta)$ iterations on a sampled training dataset $\mathcal{Z}$ using a learning rate η. Under these conditions, the following properties hold with high probability:

1. ***Standard Training:*** *The model at T iteration $F^{(T)}$ converges to a non-robust global minimum. Let F_i^T be the $F_i(X)$ at T-th iteration. In particular, there exists a perturbation $\Delta(X,y)$ independent of $F^{(T)}$ such that the robust top-1 accuracy is poor:*

$$\mathbb{P}_{(X,y)\sim\mathcal{D}}\Big[\arg\max_{i\in[k]} F_i^{(T)}(X + \Delta(X,y)) \ne y\Big] = 1 - o(1).$$

2. ***Adversarial Training:*** *The model $F^{(T)}$ converges to a robust global minimum. Consequently, for perturbations $\|\Delta\|_\infty \le \epsilon$, the robust top-1 accuracy is high:*

$$\mathbb{P}_{(X,y)\sim\mathcal{D}}\Big[\exists \Delta \in (\mathbb{R}^d)^P,\ \|\Delta\|_\infty \le \epsilon \text{ s.t. } \arg\max_{i\in[k]} F_i^{(T)}(X+\Delta) \ne y\Big]$$

Example 1 establishes that adversarial training achieves robust top-1 accuracy $1 - o(1)$, while standard training achieves only $o(1)$. This top-1 accuracy gap has a direct distributional consequence: when the model consistently identifies the correct class under attack, the nonconformity score $S^{\text{adv}} = 1 - \widehat{f}_y^{\text{adv}}(x+\delta)$ concentrates near zero, whereas standard training produces scores spread across $[0,1]$. This gives an example of (A5).

Under (A5), Theorem 3.5 and Corollary 3.6 below together derive a ratio inequality between the adversarially and cleanly-trained set-size bounds, and concludes that adversarial training yields a strictly tighter bound on the expected prediction-set size.

Theorem 3.5. *Let $P_{y,\text{true}} := \mathbb{E}[\widehat{f}_y(x_{\text{test}})]$ be the expected predictive probability assigned to the true label under P, and let Q denote the empirical β-quantile (with the standard finite-sample correction) of the HPS score $S(x,y) = 1 - \widehat{f}_y(x)$ on the calibration fold. Under* (A1)–(A4)*, Split CP returns the prediction set*

$$C(x_{\text{test}}) \;=\; \big\{\, i \in [K] : \widehat{f}_i(x_{\text{test}}) \geq 1 - Q \,\big\},$$

and its expected size satisfies

$$\mathbb{E}\big[|C(x_{\text{test}})|\big] \;\leq\; (1-\alpha) \;+\; h(\epsilon_{\text{cal}},\, \epsilon_{\text{test}}) \;+\; \frac{1 - P_{y,\text{true}}}{1 - Q}, \tag{8}$$

where $h(\epsilon_{\text{cal}}, \epsilon_{\text{test}})$ is the tolerance band of Theorem 3.4 and the last term is a distribution-free upper bound on the contribution of the $K-1$ incorrect classes, requiring no assumption on the distribution of incorrect-class probabilities.

Given the result in Theorem 3.5, we can obtain the following:

Corollary 3.6 (Adversarial training tightens the bound)**.** *Let $\widehat{f}^{\text{adv}}$ and $\widehat{f}^{\text{clean}}$ denote the adversarially- and cleanly-trained classifiers, with corresponding quantities $(Q^{\text{adv}}, P^{\text{adv}}_{y,\text{true}})$ and $(Q^{\text{clean}}, P^{\text{clean}}_{y,\text{true}})$. Under Assumption* (A5)*, Lemma B.3 gives*

$$\frac{1 - P^{\text{adv}}_{y,\text{true}}}{1 - Q^{\text{adv}}} \;\leq\; \frac{1 - P^{\text{clean}}_{y,\text{true}}}{1 - Q^{\text{clean}}},$$

so the upper bound (8) *for $\widehat{f}^{\text{adv}}$ is at most that for $\widehat{f}^{\text{clean}}$; adversarial training yields a strictly smaller bound on the expected prediction-set size.*

The proof of Theorem 3.5 and Corollary 3.6 is given in Appendix B.3. For Theorem 3.5, the argument decomposes the expected set size into two parts: the true-class contribution, which inherits the tolerance band $(1-\alpha) + h(\epsilon_{\text{cal}}, \epsilon_{\text{test}})$ from Theorem 3.4, and the contribution of the $K-1$ incorrect classes, bounded via Markov's inequality in a fully distribution-free manner.

For Corollary 3.6, intuitively, adversarial training simultaneously raises $P_{y,\text{true}}$ (higher confidence on the true class under attack) and lowers Q (smaller nonconformity scores on the calibration fold), both of which shrink the ratio $(1 - P_{y,\text{true}})/(1-Q)$ and hence the set-size bound. We validate the inequality empirically across all dataset and architecture pairs in Section C.

4. Experiments

In the experiments, we empirically validate the theoretical results presented in the theorems and propositions. Specifically, we expect that the marginal coverage of Split CP exhibits a monotonic relationship with respect to the attack strength applied to the calibration set in classification tasks for Theorem 3.2. Furthermore, for a fixed calibration-time attack strength, we demonstrate that the prediction set coverage remains within a predictable range over an interval of test-time attack strengths for Theorem 3.4. Finally, we show that adversarial training reduces the expected prediction size by increasing the probability of giving the true prediction for Theorem 3.5. The details of all the experimental results can be found in Appendix C.

4.1. Experimental Setup

Dataset CIFAR-10, CIFAR-100 (Deng et al., 2009), Tiny-ImageNet (Le and Yang, 2015), and MNIST (Deng, 2012) are used in the experiments with 60% data for training, 20% for calibration, and the remaining 20% for testing.

Neural network architecture/model We conducted image classification experiments using two distinct architectures: ResNet-50d (He et al., 2019) and Vision Transformers (ViT) (Dosovitskiy et al., 2020). Both models were trained separately under the same experimental setup, and we compared their performance to assess not only marginal coverage but also the size of the resulting conformal prediction sets. We chose to include ViT because recent work has demonstrated that transformer-based models are generally more powerful and robust than ResNet-style convolutional networks in image classification tasks under adversarial attack (Chen et al., 2021). This comparison allows us to evaluate how the choice of model architecture influences the validity and efficiency of prediction sets.

Training configuration For all datasets, we train ResNet50d and ViT to a stable training loss using 40 epochs on a single NVIDIA H200 GPU. Our evaluation focuses on ℓ_∞ adversarial attacks with perturbation strengths $8/255$ and $16/255$. We do not report ℓ_2 attacks because the clean (non-adversarially trained) models are comparatively insensitive to ℓ_2 perturbations, empirical coverage remains high, thus ℓ_2 fails to expose meaningful differences in robustness for Split CP. In contrast, ℓ_∞ attacks are stronger in our setting and reliably stress the models, providing a more stringent and discriminative robustness assessment for our experiments. We design a parameter grid with respect to $\epsilon_{train}, \epsilon_{cal}, \epsilon_{test}$ summarized in Table 1 to systematically explore empirical coverage across four different datasets, targeting a nominal coverage level of 90%.

Evaluation We evaluated empirical coverage and prediction set size across all parameter configurations and datasets. The results from clean training under each level of test-time adversarial attack serve as our baseline for comparison. All models are trained until the training loss reaches a pre-

specified target threshold, with the training process typically requiring several days to complete. Besides, while some related works, such as RSCP (Gendler et al., 2022), consider robustness of Split CP, some comparison results Fig 3 in the CIFAR-10 dataset showed that the RSCP and RSCP+ under ℓ_2 and ℓ_∞ attack on the test set failed to maintain the desired coverage and obtain a larger average set size. Therefore, we focus on comparing the performance of the approach considered in this paper.

Table 1. Adversarial training and testing configurations under different calibration levels.

Calibration ϵ	**Training / Test** ϵ	
8/255	Train: {0, 4, 8, 12, 16}/255	Test: {2–14}/255
16/255	Train: {0, 4, 8, 12, 16}/255	Test: {10–22}/255

Results To verify Theorems 3.2–3.5, we run three complementary studies. (i) For Theorem 3.2, we assess the Split CP procedure under a calibration-time attack of strength ϵ_{cal} and confirm that empirical coverage varies monotonically with ϵ_{cal}. (ii) For Theorem 3.4, fixing ϵ_{cal}, we sweep the test-time attack ϵ_{test} in a neighborhood of ϵ_{cal} (including $\epsilon_{test} = \epsilon_{cal}$ and check that coverage stays within the prescribed tolerance band around the target level. (iii) For Theorem 3.5, we compare clean training versus adversarial training while holding Split CP fixed, and show that adversarial training yields smaller expected prediction-set sizes at comparable coverage.

We begin with Theorem 3.2: we examine the monotonicity of Split CP's empirical coverage as the calibration attack strength ϵ_{cal} varies, holding the *test-time* attack range fixed across datasets. We then turn to Theorem 3.4: fixing ϵ_{cal}, we sweep ϵ_{test} to assess robustness under different evaluation conditions. Throughout, we use a tolerance $\beta = 3\%$; thus, for a 90% target, acceptable coverage lies in $[87\%, 93\%]$ over a range of ϵ_{test} provided ϵ_{cal} is chosen appropriately.

Across all models and datasets, the results align with Theorems 3.2–3.5. In Fig. 1, with adversarial training at $\epsilon_{\text{train}} = 4/255$, empirical coverage is nonincreasing in ϵ_{test} for $\epsilon_{\text{cal}} \in \{8/255, 16/255\}$; for fixed ϵ_{test}, larger ϵ_{cal} yields higher empirical coverage (Theorem 3.2). For fixed ϵ_{cal}, the ϵ_{test} interval that keeps performance within the 87%–93% band widens as ϵ_{cal} increases, while prediction-set sizes remain nearly unchanged (Theorem 3.4). Fig. 2 shows that adversarially trained models produce substantially smaller prediction sets than cleanly trained models on CIFAR-10/100, MNIST and TinyImageNet, consistent with Theorem 3.5.Additional results and the resnet50d counterparts appear in Appendix C.

We organize the evaluation around our theory. First, for Theorem 3.2, we vary the calibration attack strength ϵ_{cal} and examine the monotonicity of Split CP's empirical coverage, holding the *test-time* attack range fixed across datasets. Second, for Theorem 3.4, we fix ϵ_{cal} and sweep ϵ_{test} to assess robustness under different evaluation conditions. We use a tolerance $\beta = 3\%$; thus, for a 90% target, acceptable coverage lies in $[87\%, 93\%]$ over a range of ϵ_{test} when ϵ_{cal} is chosen appropriately.

The empirical findings are consistent with Theorems 3.2–3.5. In Fig. 1, with adversarial training at $\epsilon_{\text{train}} = 4/255$, empirical coverage is nonincreasing in ϵ_{test} for $\epsilon_{\text{cal}} \in \{8/255, 16/255\}$; at any fixed ϵ_{test}, larger ϵ_{cal} yields higher empirical coverage (Theorem 3.2). For a fixed ϵ_{cal}, the ϵ_{test}-interval that keeps performance within the 87%–93% band widens as ϵ_{cal} increases, while prediction-set sizes remain nearly unchanged (Theorem 3.4). Fig. 2 shows that adversarially trained models produce substantially smaller prediction sets than cleanly trained models on CIFAR-10/100, MNIST, and TinyImageNet (Theorem 3.5). The ResNet counterparts and additional results appear in Figs. 4–5 and Appendix C; The qualitative behavior aligns with our ViT findings: empirical coverage decreases monotonically with increasing ϵ_{test}, larger ϵ_{cal} shifts the curves upward and widens the robustness range, and adversarial training yields markedly smaller, more stable conformal set sizes.

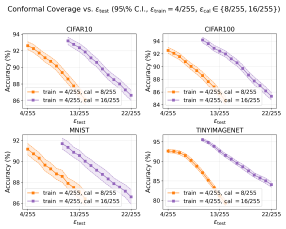

Figure 1. ViT Accuracy

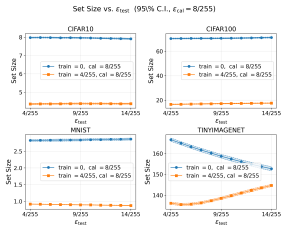

Figure 2. ViT Set Size

5. Conclusion

This work bridges the gap between conformal prediction theory and adversarial robustness, providing both theoretical insights and empirical evidence on how adversarial perturbations influence prediction set validity. By formalizing the relationship between calibration attack strength and test-time robustness, we establish conditions under which target coverage can be maintained within predefined tolerances. Our analysis further reveals that adversarial training not only enhances robustness but also reduces prediction set size through increased output entropy, thereby improving informativeness. Extensive experiments across diverse datasets and architectures confirm the monotonicity of coverage behavior, the feasibility of marginal coverage-range guarantees, and the efficiency gains from adversarial training. Collectively, these results offer a principled framework for deploying conformal prediction in adversarial environments, paving the way for reliable uncertainty quantification in safety-critical applications.

6. Related Work

6.1. Adversarial training

Adversarial examples have become central to the study of machine learning safety because of their implications for model robustness (Goodfellow et al., 2015). A growing body of research has explored the use of adversarial training as a primary defense. Despite substantial progress, modern models remain vulnerable to small, often imperceptible, perturbations in both regression and classification tasks. Recent theoretical and empirical results clarify a fundamental robustness–accuracy tradeoff in adversarially trained models. In linear settings, where analysis is more tractable, adversarial training improves robustness at the expense of standard accuracy. (Javanmard et al., 2020) precisely characterize this tradeoff and establish limits on the accuracy achievable by any algorithm. More recently, Xing et al. (Xing, 2023) showed that two-stage training with additional unlabeled data and pseudolabeling can enhance robustness even in two-layer neural networks. Taken together, these findings highlight the particular effectiveness of adversarial training for simple models, while also offering insights for extending such defenses to richer architectures. Other related literature can be found in (Goodfellow et al., 2015; Madry et al., 2018; Zhang et al., 2019; Carmon et al., 2019; Xie et al., 2020; Tanner et al., 2024; Yin et al., 2019; Javanmard and Soltanolkotabi, 2022; Dohmatob and Scetbon, 2024).

6.2. Adversarially robust conformal prediction

One critical high-stakes task in machine learning is constructing reliable prediction sets for classification tasks in the presence of adversarially perturbed test data. Recent work proposes several approaches toward this goal. (Gendler et al., 2022) demonstrated that standard CP fails to provide valid coverage under adversarial perturbations bounded in ℓ_2 norm and introduce Randomized Smoothed Conformal Prediction (RSCP), which combines random smoothing with a modified thresholding rule to improve robustness. Extending this line, (Yan et al., 2024) proposed two approaches, Post-Training Transformation (PTT) and Robust Conformal Training (RCT), that significantly reduce the size of prediction sets while maintaining valid coverage under adversarial conditions. More recently, (Scholten and Günnemann, 2025) aggregate predictions from multiple models trained on partitioned datasets, yielding provable robustness to both test-time adversarial attacks and training-time data poisoning. Complementary developments include probabilistic robustness via quantile-of-quantiles, adversarially valid guarantees for sequential settings, certifiable reasoning with probabilistic circuits, Lipschitz-bounded models for scalable certification, and conformal training/attack frameworks that directly optimize set size under threat (Bastani et al., 2022; Areces et al., 2025; Kang et al., 2024; Bao et al., 2025). Collectively, these contributions advance provably robust conformal prediction in adversarial scenarios by addressing both validity and efficiency.

Despite these progress, important gaps remain for practical deployment of robust prediction sets, in particular the limited theoretical understanding of robustness under adversarial attack procedures beyond the formulations via adversarial perturbations like in RSCP. Most existing robust CP methods are built around a fixed robustness mechanism and analyzed with adversarial perturbation. For instance, probabilistic robustness calibrated via nested quantiles (Ghosh et al., 2023), neural network verification based certification (Jeary et al., 2024), (H. Zargarbashi et al., 2024) proposed a CDF-Aware Smoothed prediction sets (CAS) a conformal prediction method that certifies coverage under both adversarial evasion and poisoning attacks, (Zargarbashi and Bojchevski, 2025) used smoothing based methods that drastically reduce the Monte Carlo burden by using binary certification, (Zargarbashi et al., 2025) including a single sample robust CP that certifies the conformal procedure rather than individual conformity scores . While these works provides strong validity guarantee under their robust model, guidance on how calibration time robustness design should be selected to obtain a desired test time coverage profile under varying attack strengths is limited. Besides, the coverage behavior induced by jointly varying calibration side and test side attack levels is typically not made explicit either. Moreover, beyond validity, efficiency has often been emphasized to evaluate the performance of the robust CP model, with tighter robust sets (H. Zargarbashi et al., 2024), lower sample certification by proposing a single-sample robust conformal prediction method that leverages a binary certificate to certify the conformal procedure(Zargarbashi et al., 2025), and scalable certification via structural constraints such as Lipschitz bounded networks (Massena et al., 2025), while set size is often treated as a byproduct of the chosen certificate rather than a controllable objective across a range of adversarial budgets. Finally, robust uncertainty quantification must often contend with distribution changes beyond bounded test time perturbations, such as strategic adaptations, which require alternative robustness models and calibration principles (Csillag et al., 2024). Motivated by these limitations, we study robust split conformal prediction under calibration time adversarial design, aiming to make adversarial coverage tractable and to clarify when robustness mechanisms can improve efficiency without sacrificing validity.

References

Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Deforges. Reveal of vision transformers robustness against adversarial attacks. *arXiv preprint arXiv:2106.03734*,

2021.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33: 16048–16059, 2020.

Felipe Areces, Christopher Mohri, Tatsunori Hashimoto, and John C. Duchi. Online conformal prediction via online optimization. In *International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=KwGc2JUIDK.

Felix Assion, Peter Schlicht, Florens Greßner, Wiebke Gunther, Fabian Huger, Nico Schmidt, and Umair Rasheed. The attack generator: A systematic approach towards constructing adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

Christopher RS Banerji, Tapabrata Chakraborti, Chris Harbron, and Ben D MacArthur. Clinical ai tools must convey predictive uncertainty for each individual patient. *Nature medicine*, 29(12):2996–2998, 2023.

Jie Bao, Chuangyin Dang, Rui Luo, Hanwei Zhang, and Zhixin Zhou. Enhancing adversarial robustness with conformal prediction: A framework for guaranteed model reliability. *arXiv preprint arXiv:2506.07804*, 2025.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Osbert Bastani, Varun Gupta, Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Practical adversarial multivalid conformal prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yoshua Bengio, Ian Goodfellow, Aaron Courville, et al. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

Emmanuel Candes, Lihua Lei, and Zhimei Ren. Conformalized survival analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(1):24–45, 2023.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pretraining or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

Daniel Csillag, Claudio José Struchiner, and Guilherme Tegoni Goedert. Strategic conformal prediction, 2024. URL https://arxiv.org/abs/2411.01596.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Elvis Dohmatob and Meyer Scetbon. Precise accuracy/robustness tradeoffs in regression: Case of general norms. In *Forty-first International Conference on Machine Learning*, 2024.

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. *arXiv preprint arXiv:2010.11929*, 2020.

Asaf Gendler, Tsui-Wei Weng, Luca Daniel, and Yaniv Romano. Adversarially robust conformal prediction. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9L1BsI4wP1H.

Subhankar Ghosh, Yuanjie Shi, Taha Belkhouja, Yan Yan, Jana Doppa, and Brian Jones. Probabilistically robust conformal prediction. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 681–690. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/ghosh23a.html.

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

Julia Grabinski, Paul Gavrikov, Janis Keuper, and Margret Keuper. Robust models are less over-confident. *Advances in Neural Information Processing Systems*, 35:39059–39075, 2022.

Yu Gui, Ying Jin, and Zhimei Ren. Conformal alignment: Knowing when to trust foundation models with guarantees. *Advances in Neural Information Processing Systems*, 37:73884–73919, 2024.

Soroush H. Zargarbashi, Mohammad Sadegh Akhondzadeh, and Aleksandar Bojchevski. Robust yet efficient conformal prediction sets. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17123–17147. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/h-zargarbashi24a.html.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019.

Jessica Hullman, Yifan Wu, Dawei Xie, Ziyang Guo, and Andrew Gelman. Conformal prediction and human decision making. *arXiv preprint arXiv:2503.11709*, 2025.

Adel Javanmard and Mohammad Mehrabi. Adversarial robustness for latent models: Revisiting the robust-standard accuracies tradeoff. *Operations Research*, 72(3):1016–1030, 2024.

Adel Javanmard and Mahdi Soltanolkotabi. Precise statistical analysis of classification accuracies for adversarial training. *The Annals of Statistics*, 50(4):2127–2156, 2022.

Adel Javanmard, Mahdi Soltanolkotabi, and Hamed Hassani. Precise tradeoffs in adversarial training for linear regression. In *Conference on Learning Theory*, pages 2034–2078. PMLR, 2020.

Linus Jeary, Tom Kuipers, Mehran Hosseini, and Nicola Paoletti. Verifiably robust conformal prediction. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=5pJfDlaSxV. NeurIPS 2024 (poster).

Mintong Kang, Nezihe Merve Gürel, Linyi Li, and Bo Li. Colep: Certifiably robust learning-reasoning conformal prediction via probabilistic circuits. *arXiv preprint arXiv:2403.11348*, 2024.

Shayan Kiyani, George Pappas, Aaron Roth, and Hamed Hassani. Decision theoretic foundations for conformal prediction: Optimal uncertainty quantification for risk-averse agents. *arXiv preprint arXiv:2502.02561*, 2025.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akilesh Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Phillips, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*, 2017.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Binghui Li and Yuanzhi Li. Adversarial training can provably improve robustness: Theoretical analysis of feature learning process under structured data. *arXiv preprint arXiv:2410.08503*, 2024.

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

Pascal Massart. The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990.

Thomas Massena, Léo Andéol, Thibaut Boissin, Franck Mamalet, Corentin Friedrich, Mathieu Serrurier, and Sébastien Gerchinovitz. Efficient robust conformal prediction via lipschitz-bounded networks, 2025. URL https://arxiv.org/abs/2506.05434.

David J Miller, Zhen Xiang, and George Kesidis. Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks. *Proceedings of the IEEE*, 108(3):402–433, 2020.

José G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *CVPR workshops*, volume 2, 2017.

Roberto I Oliveira, Paulo Orenstein, Thiago Ramos, and João Vitor Romano. Split conformal prediction and non-exchangeable data. *Journal of Machine Learning Research*, 25(225):1–38, 2024.

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

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in neural information processing systems*, 33:3581–3591, 2020.

Yan Scholten and Stephan Günnemann. Provably reliable conformal prediction sets in the presence of data poisoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ofuLWn8DFZ.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643, 2020.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

Samuel Henrique Silva and Peyman Najafirad. Opportunities and challenges in deep learning adversarial robustness: A survey. *arXiv preprint arXiv:2007.00753*, 2020.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

Kasimir Tanner, Matteo Vilucchio, Bruno Loureiro, and Florent Krzakala. A high dimensional statistical model for adversarial training: Geometry and trade-offs. *arXiv preprint arXiv:2402.05674*, 2024.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yue Xing. Adversarial training with generated data in high-dimensional regression: An asymptotic study. *arXiv preprint arXiv:2306.12582*, 2023.

Yue Xing, Qifan Song, and Guang Cheng. On the generalization properties of adversarial training. In *International Conference on Artificial Intelligence and Statistics*, pages 505–513. PMLR, 2021.

Ge Yan, Yaniv Romano, and Tsui-Wei Weng. Provably robust conformal prediction with improved efficiency. *arXiv preprint arXiv:2404.19651*, 2024.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR, 2019.

Soroush H Zargarbashi and Aleksandar Bojchevski. Robust conformal prediction with a single binary certificate. *arXiv preprint arXiv:2503.05239*, 2025.

Soroush H. Zargarbashi, Mohammad Sadegh Akhondzadeh, and Aleksandar Bojchevski. One sample is enough to make conformal prediction robust, 2025. URL https://arxiv.org/abs/2506.16553.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.

A. Comparison with RSCP and RSCP+

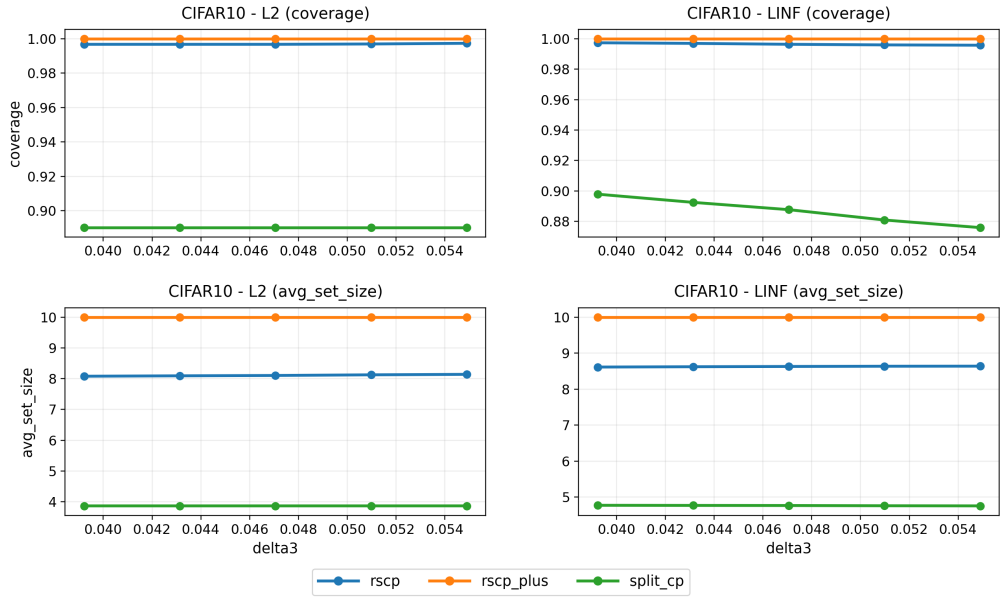

Figure 3. CIFAR-10 comparison of ℓ_2 and ℓ_∞ attacks for RSCP, RSCP+, and Split-CP, showing coverage and average prediction set size versus δ_3, with fixed non-zero perturbation settings $\delta_1 = 8/255$ and $\delta_2 = 12/255$.

B. Proofs

Lemma B.1 (Quantile shift under a small perturbation)**.** *Let $f := f_y(x) \in \mathbb{R}$ be a scalar score and let $g := \|\nabla f\| \geq 0$. Denote by F_f and p_f the CDF and PDF of f. Fix a level $\tau \in (0,1)$ and, for $\varepsilon \geq 0$, define $Z_\varepsilon := f - \varepsilon g$ and $q_\tau(\varepsilon) := Q_\tau(Z_\varepsilon)$, so that $F_{Z_\varepsilon}(q_\tau(\varepsilon)) = \tau$. Assume*

(i) *$p_f(q_\tau(0)) > 0$ and F_f is differentiable at $q_\tau(0) := Q_\tau(f)$;*

(ii) *the conditional expectation $m(x) := \mathbb{E}[g \mid f = x]$ is continuous at $x = q_\tau(0)$;*

(iii) *(f, g) has a joint density continuous in a neighbourhood of $\{x = q_\tau(0)\}$, so that differentiation under the integral sign is justified.*

Then, as $\varepsilon \to 0$,

$$Q_\tau(f - \varepsilon g) = Q_\tau(f) - \varepsilon\, \mathbb{E}[g \,|\, f = Q_\tau(f)] + o(\varepsilon). \tag{9}$$

If moreover $0 \leq g \leq L$ almost surely, then

$$\big|Q_\tau(f - \varepsilon g) - Q_\tau(f)\big| \;\leq\; \varepsilon L. \tag{10}$$

Proof of Lemma B.1. For fixed $x \in \mathbb{R}$,

$$\begin{aligned} F_{Z_\varepsilon}(x) &= \mathbb{P}(f - \varepsilon g \leq x) = \iint \mathbf{1}\{u - \varepsilon v \leq x\}\, p_{f,g}(u,v)\, du\, dv \\ &= \int \left(\int_{-\infty}^{x + \varepsilon v} p_{f,g}(u,v)\, du \right) dv. \end{aligned}$$

Assumption (iii) lets differentiation in ε pass under the integral; at $\varepsilon = 0$,

$$\frac{\partial}{\partial \varepsilon} F_{Z_\varepsilon}(x)\Big|_{\varepsilon=0} = \int v\, p_{f,g}(x,v)\, dv = p_f(x)\, \mathbb{E}[g \mid f = x]. \tag{11}$$

Since $F_{Z_\varepsilon}(q_\tau(\varepsilon)) = \tau$ for all ε, the chain rule at $\varepsilon = 0$ combined with (11) gives

$$0 = p_f(q_\tau(0))\, \mathbb{E}[g \mid f = q_\tau(0)] + p_f(q_\tau(0))\, q'_\tau(0),$$

so that $q'_\tau(0) = -\mathbb{E}[g \mid f = q_\tau(0)]$. A first-order Taylor expansion of q_τ at $\varepsilon = 0$ yields (9). If $0 \le g \le L$ almost surely, the sandwich $f - \varepsilon L \le f - \varepsilon g \le f$ together with monotonicity of quantiles gives (10). □

Lemma B.2 (Sample quantile approximation)**.** *Let $Z_1, \dots, Z_n$ be i.i.d. with CDF F and density g_Z continuous and strictly positive at $Q_\alpha(Z)$. Set $\beta = \lceil (n+1)\alpha \rceil / n$ and let $\hat{Q}_\beta = \hat{Q}_\beta(Z_1, \dots, Z_n)$ denote the empirical β-quantile. Then*

$$\hat{Q}_\beta - Q_\alpha(Z) \;=\; O_p(n^{-1/2}).$$

More precisely, for any $\delta \in (0,1)$, with probability at least $1 - \delta$,

$$\left|\hat{Q}_\beta - Q_\alpha(Z)\right| \;\le\; \frac{\sqrt{2\log(2/\delta)}}{g_Z(Q_\alpha(Z))\,\sqrt{n}} \;+\; O(n^{-1}),$$

where the $O(n^{-1})$ term absorbs the β-to-α correction $|Q_\beta(Z) - Q_\alpha(Z)| = O(n^{-1})$.

Proof. Decompose

$$\hat{Q}_\beta - Q_\alpha \;=\; \underbrace{(\hat{Q}_\beta - Q_\beta)}_{\text{(I)}} + \underbrace{(Q_\beta - Q_\alpha)}_{\text{(II)}}.$$

Term (II). Since $\beta = \lceil (n+1)\alpha \rceil / n$ we have $|\beta - \alpha| \le 1/n$. By the local Lipschitz property of quantiles under positive density,

$$|Q_\beta(Z) - Q_\alpha(Z)| \;\le\; \frac{|\beta - \alpha|}{g_Z(Q_\alpha)} + o(|\beta - \alpha|) \;=\; O(n^{-1}).$$

Term (I). Let F_n denote the empirical CDF of $Z_1, \dots, Z_n$. By the Dvoretzky–Kiefer–Wolfowitz inequality (Massart, 1990),

$$\mathbb{P}\left(\sup_x |F_n(x) - F(x)| > \epsilon\right) \;\le\; 2e^{-2n\epsilon^2}.$$

Setting $\epsilon = \sqrt{\log(2/\delta)/(2n)}$ gives, with probability at least $1 - \delta$, the event $\mathcal{E} := \{\|F_n - F\|_\infty \le \epsilon\}$.

On $\mathcal{E}$, suppose $\hat{Q}_\beta - Q_\beta > t > 0$. Then $F_n(Q_\beta + t) < \beta$, so $F(Q_\beta + t) < \beta + \epsilon$. But $F(Q_\beta + t) \ge F(Q_\beta) + g_Z(Q_\beta)\, t + o(t) = \beta + g_Z(Q_\beta)\, t + o(t)$, giving $g_Z(Q_\beta)\, t \le \epsilon + o(t)$, hence $t \le \epsilon / g_Z(Q_\beta) + o(1)$. An identical argument handles $t < 0$. Therefore on $\mathcal{E}$,

$$|\hat{Q}_\beta - Q_\beta| \;\le\; \frac{\epsilon}{g_Z(Q_\beta)} \;=\; \frac{\sqrt{\log(2/\delta)/(2n)}}{g_Z(Q_\alpha) + o(1)}.$$

Combining terms (I) and (II) and absorbing $o(1)$ in g_Z into the constant completes the proof. □

Lemma B.3 (A5 implies the ratio inequality)**.** *Under A5,*

$$\frac{1 - P^{\text{adv}}_{y,\text{true}}}{1 - Q^{\text{adv}}} \;\le\; \frac{1 - P^{\text{clean}}_{y,\text{true}}}{1 - Q^{\text{clean}}}.$$

Proof. Since $S^\star = 1 - \widehat{f}^\star_y$, we have

$$P^\star_{y,\text{true}} \;=\; \mathbb{E}\Big[\widehat{f}^\star_y(X)\Big] \;=\; 1 - \mathbb{E}[S^\star], \qquad Q^\star \;=\; F^{-1}_{S^\star}(\beta),$$

where $\star \in \{\mathrm{adv}, \mathrm{clean}\}$ and β is the conformal quantile level.

By A5, S^{adv} is first-order stochastically smaller than S^{clean}, i.e.,

$$F_{S^{\mathrm{adv}}}(t) \;\geq\; F_{S^{\mathrm{clean}}}(t) \qquad \text{for all } t \in \mathbb{R}.$$

This yields two consequences.

Consequence 1 (numerator). First-order stochastic dominance implies $\mathbb{E}[S^{\mathrm{adv}}] \leq \mathbb{E}[S^{\mathrm{clean}}]$, hence

$$1 - P^{\mathrm{adv}}_{y,\mathrm{true}} \;\leq\; 1 - P^{\mathrm{clean}}_{y,\mathrm{true}}. \tag{12}$$

Consequence 2 (denominator). Monotonicity of quantiles under stochastic dominance gives

$$Q^{\mathrm{adv}} \;=\; F^{-1}_{S^{\mathrm{adv}}}(\beta) \;\leq\; F^{-1}_{S^{\mathrm{clean}}}(\beta) \;=\; Q^{\mathrm{clean}},$$

so that

$$1 - Q^{\mathrm{adv}} \;\geq\; 1 - Q^{\mathrm{clean}} \;>\; 0, \tag{13}$$

where the strict positivity follows from Assumption (A4).

Combining. Applying (12) to the numerator and (13) to the denominator,

$$\frac{1 - P^{\mathrm{adv}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{adv}}} \;\leq\; \frac{1 - P^{\mathrm{clean}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{adv}}} \;\leq\; \frac{1 - P^{\mathrm{clean}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{clean}}},$$

which completes the proof. □

B.1. Proof of Theorem 3.2

Proof of Theorem 3.2. We use nonconformity score $S(x, y) = 1 - \widehat{f}_y(x)$.The Split CP prediction set is $C_{\epsilon_{\mathrm{cal}}}(x) = \{y : S(x, y) \leq \hat{q}\}$ with empirical threshold

$$\hat{q} = Q_{1-\beta}(\{S(x_i + \epsilon_{\mathrm{cal}} A_i, y_i)\}_{i \in I_2}), \qquad \beta = \frac{\lceil (|I_2| + 1)\alpha \rceil}{|I_2|}.$$

Since $Q_{1-\beta}(1 - Y) = 1 - Q_\beta(Y)$, the coverage event $\{S(\tilde{x}_{\mathrm{test}}, y_{\mathrm{test}}) \leq \hat{q}\}$ is equivalent to

$$\widehat{f}_{y_{\mathrm{test}}}(\tilde{x}_{\mathrm{test}}) \;\geq\; \hat{q}_\beta(\epsilon_{\mathrm{cal}}), \qquad \hat{q}_\beta(\epsilon_{\mathrm{cal}}) := Q_\beta\Big(\{\widehat{f}_{y_i}(x_i + \epsilon_{\mathrm{cal}} A_i)\}_{i \in I_2}\Big). \tag{14}$$

Error terms from Proposition 3.1. Throughout this proof we use the quantities e_{train}, d_{cal}, and e_{cal} exactly as defined in Proposition 3.1; We also write p_f for the PDF of $\widehat{f}_y(x)$ under P (denoted g in the theorem statement) to avoid a collision with the gradient norm $\|\nabla \widehat{f}_y\|$ appearing in Lemma B.1.

Test-side expansion. By the definition $b_A(x, y) := -A(x, y)^\top \nabla_x \widehat{f}_y(x)$ and a first-order Taylor expansion under (A2)–(A3),

$$\widehat{f}_{y_{\mathrm{test}}}(\tilde{x}_{\mathrm{test}}) = \widehat{f}_{y_{\mathrm{test}}}(x_{\mathrm{test}}) - \epsilon_{\mathrm{test}}\, b_A(x_{\mathrm{test}}, y_{\mathrm{test}}) + r_{\mathrm{test}}, \qquad r_{\mathrm{test}} = O(\epsilon^2_{\mathrm{test}}). \tag{15}$$

Calibration-side population quantile. Define the deterministic population α-quantile of the adversarially perturbed calibration scores:

$$q_\alpha(\epsilon) := Q_\alpha\Big(\widehat{f}_{y_{\mathrm{cal}}}(x_{\mathrm{cal}}) - \epsilon\, b_A(x_{\mathrm{cal}}, y_{\mathrm{cal}})\Big).$$

Applying Lemma B.1 at level $\tau = \alpha$ (with $g_{\mathrm{Lemma}} = b_A$ under (A2)–(A3)),

$$q_\alpha(\epsilon_{\mathrm{cal}}) = Q_\alpha(\widehat{f}_{y_{\mathrm{cal}}}(x_{\mathrm{cal}})) - \epsilon_{\mathrm{cal}}\, c_A + \rho, \tag{16}$$

where c_A agrees with the theorem statement by exchangeability, and $|\rho| \le e_{\text{cal}}$ by definition of e_{cal} in Proposition 3.1.

Empirical-to-population gap. Apply Lemma B.2 to $\{\widehat{f}_{y_i}(x_i + \epsilon_{\text{cal}} A_i)\}_{i \in I_2}$:

$$\hat{q}_\beta(\epsilon_{\text{cal}}) = q_\alpha(\epsilon_{\text{cal}}) + \Delta_n, \qquad |\Delta_n| = O_p(|I_2|^{-1/2}). \tag{17}$$

By Proposition 3.1, $p_f(Q_\alpha)\,|\mathbb{E}[\Delta_n]| \le d_{\text{cal}}$.

Auxiliary function. Define

$$h(\varepsilon, t) := \mathbb{P}\Big(\widehat{f}_{y_{\text{test}}}(x_{\text{test}}) - \varepsilon\, b_A(x_{\text{test}}, y_{\text{test}}) \ge t\Big),$$

where the probability is over $(x_{\text{test}}, y_{\text{test}})$ only and t is deterministic. Conditioning on I_2, substituting (15) into (14), and using $|\partial_t h| \le p_f(Q_\alpha) < \infty$ to absorb $r_{\text{test}} = O(\epsilon_{\text{test}}^2)$,

$$\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(\tilde{x}_{\text{test}}) \mid I_2) = h(\epsilon_{\text{test}},\ \hat{q}_\beta(\epsilon_{\text{cal}})) + O(\epsilon_{\text{test}}^2). \tag{18}$$

Step 1: empirical → population threshold. First-order Taylor in t around $q_\alpha(\epsilon_{\text{cal}})$ and (17):

$$h(\epsilon_{\text{test}},\ \hat{q}_\beta(\epsilon_{\text{cal}})) = h(\epsilon_{\text{test}},\ q_\alpha(\epsilon_{\text{cal}})) - p_f(Q_\alpha)\,\Delta_n + O(\Delta_n^2). \tag{19}$$

Step 2: population perturbed → unperturbed quantile. First-order Taylor in t around $Q_\alpha - \epsilon_{\text{cal}} c_A$ and (16):

$$h(\epsilon_{\text{test}},\ q_\alpha(\epsilon_{\text{cal}})) = h(\epsilon_{\text{test}},\ Q_\alpha - \epsilon_{\text{cal}}\, c_A) - p_f(Q_\alpha)\,\rho + O(\rho^2). \tag{20}$$

Taylor expansion of h. Let $p_{f,b_A}(t, s)$ be the joint density of $(\widehat{f}_y(x), b_A(x, y))$ under P. By (A3) and dominated convergence,

$$\partial_\varepsilon h(\varepsilon, t)\big|_{\varepsilon=0} = -p_f(t)\,\mathbb{E}[b_A(x, y) \mid \widehat{f}_y(x) = t],$$

so $\partial_\varepsilon h(0, Q_\alpha) = -c_A\, p_f(Q_\alpha)$ and $\partial_t h(0, Q_\alpha) = -p_f(Q_\alpha)$. Since $\widehat{f}_{y_{\text{test}}}(x_{\text{test}})$ and $\widehat{f}_{y_{\text{cal}}}(x_{\text{cal}})$ are identically distributed and the CDF is continuous at Q_α,

$$h(0, Q_\alpha) = \mathbb{P}\Big(\widehat{f}_{y_{\text{test}}}(x_{\text{test}}) \ge Q_\alpha\Big(\widehat{f}_{y_{\text{cal}}}(x_{\text{cal}})\Big)\Big) = 1 - \alpha.$$

A joint first-order Taylor expansion of h at $(0, Q_\alpha)$ gives

$$h(\epsilon_{\text{test}},\, Q_\alpha - \epsilon_{\text{cal}}\, c_A) = (1 - \alpha) - (\epsilon_{\text{test}} - \epsilon_{\text{cal}})\, c_A\, p_f(Q_\alpha) + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2). \tag{21}$$

Assembly. Combining (18)–(21),

$$\begin{aligned}\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(\tilde{x}_{\text{test}}) \mid I_2) = (1 - \alpha) &- (\epsilon_{\text{test}} - \epsilon_{\text{cal}})\, c_A\, p_f(Q_\alpha) \\ &- p_f(Q_\alpha)\,\Delta_n - p_f(Q_\alpha)\,\rho + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2 + \Delta_n^2 + \rho^2).\end{aligned} \tag{22}$$

Taking expectation over I_2. Apply $\mathbb{E}_{I_2}$ to (22). The term $p_f(Q_\alpha)\,\Delta_n$ has $\mathbb{E}[p_f(Q_\alpha)\,\Delta_n] = p_f(Q_\alpha)\,\mathbb{E}[\Delta_n]$,and by Proposition 3.1 $p_f(Q_\alpha)\,|\mathbb{E}[\Delta_n]| \le d_{\text{cal}}$; the $O_p(|I_2|^{-1/2})$ fluctuation of Δ_n is zero-mean and disappears under expectation. The term $p_f(Q_\alpha)\,|\rho| \le e_{\text{cal}}$ by Proposition 3.1.The classifier approximation error e_{train} enters via the $O(\cdot)$ remainder from Proposition 3.1.Thus,

$$\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(\tilde{x}_{\text{test}})) = (1 - \alpha) - (\epsilon_{\text{test}} - \epsilon_{\text{cal}})\, c_A\, p_f(Q_\alpha) + \mathcal{E},$$

where $|\mathcal{E}| \le e_{\text{train}} + d_{\text{cal}} + e_{\text{cal}} + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2)$, as claimed. □

B.2. Proof of Theorem 3.4

Proof of Theorem 3.4. By Theorem 3.2, the signed coverage gap satisfies

$$\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(x_{\text{test}} + \epsilon_{\text{test}} A_{\text{test}})) - (1-\alpha) \;=\; -(\epsilon_{\text{test}} - \epsilon_{\text{cal}})\, c\, g(Q_\alpha(f_{y_{\text{cal}}})) + E + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2), \tag{23}$$

with a single residual $E = e_{\text{train}} + d_{\text{cal}} + e_{\text{cal}}$ obtained from Proposition 3.1. Applying the triangle inequality to (23),

$$|\mathbb{P}(\cdot) - (1-\alpha)| \;\le\; |\epsilon_{\text{test}} - \epsilon_{\text{cal}}|\, c\, g(Q_\alpha) + E + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2).$$

For the guarantee $|\mathbb{P}(\cdot) - (1-\alpha)| \le \tau$ to hold for every admissible realization of R (up to the $O(\epsilon^2)$ remainder), it suffices that

$$|\epsilon_{\text{test}} - \epsilon_{\text{cal}}|\, c\, g(Q_\alpha) + E \;\le\; \tau,$$

which requires $\tau > E$ for a non-empty admissible region.Dividing by $c\, g(Q_\alpha) > 0$,

$$|\epsilon_{\text{test}} - \epsilon_{\text{cal}}| \;\le\; \frac{\tau - E}{c\, g(Q_\alpha(f_{y_{\text{cal}}}))}, \tag{24}$$

which is exactly the symmetric interval about ϵ_{cal} stated in the theorem. The interval is centered at ϵ_{cal} because the adversarial coefficient in (23) is linear in $\epsilon_{\text{test}} - \epsilon_{\text{cal}}$ and vanishes at the matched budget, so coverage deviates from $1-\alpha$ only by $E + O(\epsilon^2)$ when $\epsilon_{\text{test}} = \epsilon_{\text{cal}}$, which always lies inside the tolerance window whenever $\tau > E$. □

B.3. Proof of Theorem 3.5

Proof of Theorem 3.5. Decompose the expected prediction-set size by separating the true class from the $K-1$ incorrect classes:

$$\mathbb{E}\big[|C(x_{\text{test}})|\big] = \mathbb{P}(y_{\text{test}} \in C(x_{\text{test}})) + \mathbb{E}\left[\sum_{i \neq y_{\text{test}}} \mathbb{I}\Big\{\widehat{f}_i(x_{\text{test}}) \ge 1 - Q\Big\}\right]. \tag{25}$$

True-class term. By Theorem 3.4,

$$|\mathbb{P}(y_{\text{test}} \in C_{\epsilon_{\text{cal}}}(x_{\text{test}} + \epsilon_{\text{test}} A_{\text{test}})) - (1-\alpha)| \;\le\; h(\epsilon_{\text{cal}}, \epsilon_{\text{test}}) + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2),$$

so that

$$\mathbb{P}(y_{\text{test}} \in C(x_{\text{test}})) \;\le\; (1-\alpha) + h(\epsilon_{\text{cal}}, \epsilon_{\text{test}}) + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2). \tag{26}$$

Incorrect-class term. For each fixed label i, the output $\widehat{f}_i(x_{\text{test}}) \ge 0$ and $1 - Q > 0$ by Assumption (A4). Applying Markov's inequality conditionally on y_{test} and then taking expectation via the tower property,

$$\mathbb{E}\left[\sum_{i \neq y_{\text{test}}} \mathbb{I}\Big\{\widehat{f}_i(x_{\text{test}}) \ge 1 - Q\Big\}\right] \;\le\; \mathbb{E}\left[\sum_{i \neq y_{\text{test}}} \frac{\widehat{f}_i(x_{\text{test}})}{1-Q}\right] \;=\; \frac{\mathbb{E}\Big[\sum_{i \neq y_{\text{test}}} \widehat{f}_i(x_{\text{test}})\Big]}{1-Q}.$$

Since $\sum_{i=1}^{K} \widehat{f}_i(x_{\text{test}}) = 1$ P-almost surely, we have $\sum_{i \neq y_{\text{test}}} \widehat{f}_i(x_{\text{test}}) = 1 - \widehat{f}_{y_{\text{test}}}(x_{\text{test}})$, and therefore

$$\mathbb{E}\left[\sum_{i \neq y_{\text{test}}} \mathbb{I}\Big\{\widehat{f}_i(x_{\text{test}}) \ge 1 - Q\Big\}\right] \;\le\; \frac{1 - P_{y,\text{true}}}{1-Q}. \tag{27}$$

Combining. Substituting (26) and (27) into (25) gives

$$\mathbb{E}\big[|C(x_{\text{test}})|\big] \;\le\; (1-\alpha) + h(\epsilon_{\text{cal}}, \epsilon_{\text{test}}) + \frac{1 - P_{y,\text{true}}}{1-Q} + O(\epsilon_{\text{cal}}^2 + \epsilon_{\text{test}}^2),$$

which is the bound (8).

Adversarial training comparison. Applying bound (8) to both classifiers,

$$\mathbb{E}\big[|C^{\mathrm{adv}}(x_{\mathrm{test}})|\big] \;\le\; (1-\alpha) + h(\epsilon_{\mathrm{cal}}, \epsilon_{\mathrm{test}}) + \frac{1 - P^{\mathrm{adv}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{adv}}},$$

$$\mathbb{E}\big[|C^{\mathrm{clean}}(x_{\mathrm{test}})|\big] \;\le\; (1-\alpha) + h(\epsilon_{\mathrm{cal}}, \epsilon_{\mathrm{test}}) + \frac{1 - P^{\mathrm{clean}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{clean}}}.$$

By Lemma B.3, Assumption (B) implies

$$\frac{1 - P^{\mathrm{adv}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{adv}}} \;\le\; \frac{1 - P^{\mathrm{clean}}_{y,\mathrm{true}}}{1 - Q^{\mathrm{clean}}},$$

so the upper bound for $\widehat{f}^{\mathrm{adv}}$ is at most that for $\widehat{f}^{\mathrm{clean}}$. □

C. Experiments results

C.1. Numerical results

Table 2. ViT-MAE results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	92.976	93.451	92.501	7.965
0.000	0.031	0.012	92.479	92.968	91.990	7.973
0.000	0.031	0.016	91.943	92.442	91.445	7.978
0.000	0.031	0.020	91.189	91.671	90.707	7.982
0.000	0.031	0.024	90.815	91.314	90.316	7.979
0.000	0.031	0.027	89.940	90.477	89.403	7.969
0.000	0.031	0.031	89.298	89.848	88.748	7.970
0.000	0.031	0.035	88.683	89.266	88.099	7.964
0.000	0.031	0.039	88.192	88.809	87.576	7.956
0.000	0.031	0.043	87.279	87.852	86.706	7.943
0.000	0.031	0.047	86.583	87.198	85.968	7.933
0.000	0.031	0.051	86.044	86.703	85.384	7.923
0.000	0.031	0.055	85.193	85.825	84.560	7.908
0.000	0.063	0.039	93.128	93.589	92.668	8.622
0.000	0.063	0.043	92.666	93.172	92.161	8.617
0.000	0.063	0.047	92.045	92.539	91.550	8.605
0.000	0.063	0.051	91.484	91.988	90.980	8.596
0.000	0.063	0.055	90.904	91.415	90.394	8.582
0.000	0.063	0.059	90.182	90.746	89.619	8.572
0.000	0.063	0.063	89.588	90.123	89.053	8.556
0.000	0.063	0.067	88.952	89.545	88.359	8.546
0.000	0.063	0.071	88.432	89.019	87.845	8.532
0.000	0.063	0.075	87.857	88.424	87.290	8.518
0.000	0.063	0.078	87.290	87.843	86.736	8.505
0.000	0.063	0.082	86.670	87.245	86.095	8.495
0.000	0.063	0.086	86.014	86.613	85.414	8.483
0.016	0.031	0.008	93.645	94.035	93.254	4.340
0.016	0.031	0.012	93.166	93.604	92.728	4.347
0.016	0.031	0.016	92.620	93.096	92.144	4.356
0.016	0.031	0.020	92.274	92.765	91.784	4.364
0.016	0.031	0.024	91.737	92.214	91.261	4.366
0.016	0.031	0.027	91.169	91.694	90.645	4.369
0.016	0.031	0.031	90.732	91.260	90.204	4.376
0.016	0.031	0.035	90.176	90.720	89.632	4.381
0.016	0.031	0.039	89.380	89.965	88.795	4.376
0.016	0.031	0.043	88.602	89.174	88.030	4.378
0.016	0.031	0.047	87.768	88.389	87.146	4.375
0.016	0.031	0.051	86.955	87.552	86.358	4.370
0.016	0.031	0.055	86.115	86.735	85.495	4.373
0.016	0.063	0.039	93.690	94.129	93.251	5.311
0.016	0.063	0.043	93.193	93.635	92.751	5.309
0.016	0.063	0.047	92.771	93.227	92.316	5.304
0.016	0.063	0.051	92.376	92.875	91.877	5.306
0.016	0.063	0.055	91.863	92.373	91.353	5.304
0.016	0.063	0.059	91.150	91.712	90.588	5.299
0.016	0.063	0.063	90.604	91.123	90.085	5.290
0.016	0.063	0.067	89.823	90.372	89.273	5.279
0.016	0.063	0.071	89.097	89.651	88.543	5.264
0.016	0.063	0.075	88.522	89.088	87.956	5.257
0.016	0.063	0.078	88.008	88.583	87.432	5.244
0.016	0.063	0.082	87.289	87.907	86.672	5.242
0.016	0.063	0.086	86.639	87.245	86.034	5.233
0.031	0.031	0.008	91.298	91.785	90.810	3.011
0.031	0.031	0.012	91.402	91.871	90.933	2.925
0.031	0.031	0.016	91.490	92.065	90.915	2.844

Continued on next page

Table 2. ViT-MAE results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.031	0.031	0.020	91.680	92.196	91.164	2.782
0.031	0.031	0.024	91.512	92.011	91.012	2.745
0.031	0.031	0.027	91.516	92.062	90.971	2.720
0.031	0.031	0.031	91.425	91.941	90.909	2.707
0.031	0.031	0.035	91.130	91.636	90.624	2.725
0.031	0.031	0.039	90.816	91.319	90.314	2.757
0.031	0.031	0.043	90.644	91.118	90.171	2.805
0.031	0.031	0.047	90.466	90.938	89.993	2.858
0.031	0.031	0.051	90.205	90.752	89.659	2.913
0.031	0.031	0.055	89.921	90.456	89.386	2.978
0.031	0.063	0.039	91.549	92.084	91.014	2.877
0.031	0.063	0.043	91.422	91.916	90.929	2.925
0.031	0.063	0.047	91.250	91.797	90.704	2.975
0.031	0.063	0.051	90.863	91.346	90.381	3.040
0.031	0.063	0.055	90.667	91.199	90.136	3.107
0.031	0.063	0.059	90.297	90.854	89.740	3.172
0.031	0.063	0.063	90.065	90.561	89.568	3.248
0.031	0.063	0.067	89.738	90.285	89.191	3.323
0.031	0.063	0.071	89.454	89.991	88.917	3.398
0.031	0.063	0.075	89.217	89.790	88.644	3.474
0.031	0.063	0.078	89.172	89.695	88.649	3.543
0.031	0.063	0.082	88.861	89.473	88.249	3.607
0.031	0.063	0.086	88.559	89.183	87.935	3.674
0.047	0.031	0.008	70.707	71.523	69.891	1.211
0.047	0.031	0.012	77.960	78.649	77.270	1.187
0.047	0.031	0.016	82.620	83.332	81.908	1.161
0.047	0.031	0.020	85.684	86.377	84.991	1.141
0.047	0.031	0.024	87.944	88.516	87.372	1.127
0.047	0.031	0.027	89.186	89.808	88.564	1.121
0.047	0.031	0.031	89.950	90.451	89.448	1.112
0.047	0.031	0.035	90.780	91.278	90.282	1.105
0.047	0.031	0.039	91.231	91.685	90.778	1.104
0.047	0.031	0.043	91.581	92.063	91.099	1.105
0.047	0.031	0.047	91.788	92.305	91.271	1.105
0.047	0.031	0.051	91.904	92.418	91.390	1.106
0.047	0.031	0.055	91.958	92.464	91.451	1.105
0.047	0.063	0.039	88.800	89.319	88.281	1.029
0.047	0.063	0.043	89.387	90.007	88.767	1.033
0.047	0.063	0.047	89.680	90.274	89.087	1.034
0.047	0.063	0.051	89.779	90.341	89.217	1.031
0.047	0.063	0.055	89.836	90.351	89.321	1.031
0.047	0.063	0.059	89.897	90.432	89.361	1.030
0.047	0.063	0.063	89.947	90.429	89.465	1.030
0.047	0.063	0.067	89.939	90.485	89.394	1.032
0.047	0.063	0.071	89.645	90.185	89.106	1.029
0.047	0.063	0.075	89.330	89.904	88.755	1.028
0.047	0.063	0.078	89.154	89.683	88.626	1.030
0.047	0.063	0.082	88.899	89.389	88.410	1.032
0.047	0.063	0.086	88.471	89.063	87.880	1.031
0.063	0.031	0.008	67.145	68.012	66.279	1.846
0.063	0.031	0.012	74.824	75.520	74.128	1.759
0.063	0.031	0.016	80.112	80.874	79.349	1.683
0.063	0.031	0.020	84.514	85.184	83.844	1.617
0.063	0.031	0.024	87.071	87.699	86.443	1.559
0.063	0.031	0.027	88.911	89.488	88.335	1.507
0.063	0.031	0.031	90.295	90.782	89.808	1.457
0.063	0.031	0.035	91.342	91.837	90.846	1.423
0.063	0.031	0.039	92.192	92.679	91.704	1.398
0.063	0.031	0.043	92.871	93.342	92.401	1.372

Continued on next page

Table 2. ViT-MAE results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.063	0.031	0.047	93.348	93.790	92.906	1.351
0.063	0.031	0.051	93.822	94.283	93.361	1.342
0.063	0.031	0.055	94.125	94.558	93.692	1.330
0.063	0.063	0.039	87.117	87.720	86.514	1.106
0.063	0.063	0.043	87.967	88.509	87.426	1.098
0.063	0.063	0.047	88.450	89.052	87.849	1.090
0.063	0.063	0.051	88.929	89.497	88.362	1.084
0.063	0.063	0.055	89.202	89.809	88.595	1.079
0.063	0.063	0.059	89.575	90.102	89.049	1.075
0.063	0.063	0.063	90.030	90.559	89.501	1.076
0.063	0.063	0.067	90.379	90.863	89.896	1.075
0.063	0.063	0.071	90.518	91.049	89.987	1.072
0.063	0.063	0.075	90.512	91.072	89.953	1.071
0.063	0.063	0.078	90.536	91.037	90.035	1.070
0.063	0.063	0.082	90.546	91.058	90.034	1.068
0.063	0.063	0.086	90.490	91.006	89.975	1.069

Table 3. ViT-MAE results on CIFAR100.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	95.982	96.336	95.628	70.129
0.000	0.031	0.012	95.229	95.599	94.858	70.285
0.000	0.031	0.016	94.504	94.929	94.079	70.342
0.000	0.031	0.020	93.707	94.135	93.280	70.396
0.000	0.031	0.024	92.693	93.202	92.184	70.467
0.000	0.031	0.027	91.918	92.447	91.390	70.492
0.000	0.031	0.031	91.145	91.639	90.650	70.497
0.000	0.031	0.035	90.329	90.862	89.796	70.571
0.000	0.031	0.039	89.252	89.778	88.725	70.634
0.000	0.031	0.043	88.497	89.108	87.886	70.759
0.000	0.031	0.047	87.701	88.289	87.113	70.878
0.000	0.031	0.051	86.928	87.560	86.296	71.046
0.000	0.031	0.055	86.447	87.044	85.849	71.190
0.000	0.063	0.039	93.665	94.072	93.258	79.742
0.000	0.063	0.043	93.116	93.576	92.655	79.823
0.000	0.063	0.047	92.534	93.053	92.016	79.947
0.000	0.063	0.051	91.857	92.380	91.333	80.057
0.000	0.063	0.055	91.219	91.711	90.727	80.186
0.000	0.063	0.059	90.759	91.251	90.267	80.373
0.000	0.063	0.063	90.613	91.130	90.096	80.585
0.000	0.063	0.067	90.147	90.718	89.576	80.735
0.000	0.063	0.071	90.037	90.604	89.471	80.944
0.000	0.063	0.075	89.694	90.264	89.125	81.143
0.000	0.063	0.078	89.392	89.993	88.790	81.323
0.000	0.063	0.082	89.131	89.697	88.564	81.538
0.000	0.063	0.086	89.053	89.629	88.477	81.737
0.016	0.031	0.008	93.247	93.709	92.785	16.332
0.016	0.031	0.012	92.894	93.353	92.435	16.445
0.016	0.031	0.016	92.540	93.003	92.077	16.574
0.016	0.031	0.020	92.082	92.608	91.556	16.681
0.016	0.031	0.024	91.566	92.072	91.059	16.833
0.016	0.031	0.027	90.781	91.251	90.311	16.935
0.016	0.031	0.031	90.025	90.537	89.513	17.062
0.016	0.031	0.035	89.360	89.930	88.790	17.183
0.016	0.031	0.039	88.584	89.158	88.009	17.272
0.016	0.031	0.043	87.633	88.271	86.994	17.339
0.016	0.031	0.047	86.791	87.461	86.120	17.452

Continued on next page

Table 3. ViT-MAE results on CIFAR100.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.016	0.031	0.051	85.775	86.374	85.176	17.528
0.016	0.031	0.055	84.637	85.285	83.989	17.586
0.016	0.063	0.039	94.179	94.612	93.745	30.805
0.016	0.063	0.043	93.605	94.040	93.170	31.002
0.016	0.063	0.047	92.878	93.344	92.412	31.175
0.016	0.063	0.051	92.433	92.919	91.947	31.330
0.016	0.063	0.055	91.984	92.477	91.491	31.435
0.016	0.063	0.059	91.228	91.760	90.696	31.575
0.016	0.063	0.063	90.597	91.113	90.082	31.612
0.016	0.063	0.067	89.706	90.242	89.169	31.720
0.016	0.063	0.071	88.683	89.232	88.134	31.757
0.016	0.063	0.075	87.740	88.360	87.120	31.838
0.016	0.063	0.078	87.007	87.595	86.420	31.863
0.016	0.063	0.082	86.172	86.805	85.539	31.947
0.016	0.063	0.086	85.338	85.912	84.763	31.978
0.031	0.031	0.008	84.717	85.341	84.094	5.351
0.031	0.031	0.012	86.503	87.156	85.851	5.193
0.031	0.031	0.016	87.528	88.112	86.945	5.078
0.031	0.031	0.020	88.762	89.335	88.190	4.994
0.031	0.031	0.024	89.133	89.700	88.566	4.940
0.031	0.031	0.027	89.387	89.946	88.828	4.892
0.031	0.031	0.031	89.737	90.253	89.221	4.897
0.031	0.031	0.035	89.808	90.387	89.230	4.913
0.031	0.031	0.039	89.887	90.426	89.348	4.939
0.031	0.031	0.043	89.976	90.485	89.467	4.975
0.031	0.031	0.047	89.732	90.241	89.222	5.034
0.031	0.031	0.051	89.407	89.948	88.866	5.098
0.031	0.031	0.055	89.038	89.570	88.507	5.181
0.031	0.063	0.039	91.170	91.697	90.644	5.725
0.031	0.063	0.043	91.025	91.587	90.463	5.790
0.031	0.063	0.047	90.998	91.524	90.472	5.868
0.031	0.063	0.051	90.662	91.178	90.146	5.948
0.031	0.063	0.055	90.412	90.915	89.908	6.036
0.031	0.063	0.059	89.944	90.530	89.357	6.125
0.031	0.063	0.063	89.548	90.059	89.036	6.218
0.031	0.063	0.067	89.121	89.690	88.552	6.326
0.031	0.063	0.071	88.667	89.205	88.129	6.435
0.031	0.063	0.075	88.168	88.775	87.562	6.540
0.031	0.063	0.078	87.676	88.284	87.067	6.632
0.031	0.063	0.082	87.060	87.697	86.423	6.752
0.031	0.063	0.086	86.499	87.132	85.866	6.851
0.047	0.031	0.008	82.587	83.261	81.913	3.885
0.047	0.031	0.012	85.374	86.065	84.684	3.725
0.047	0.031	0.016	87.279	87.911	86.646	3.570
0.047	0.031	0.020	88.925	89.486	88.363	3.419
0.047	0.031	0.024	90.053	90.616	89.489	3.300
0.047	0.031	0.027	90.802	91.360	90.243	3.198
0.047	0.031	0.031	91.408	91.884	90.933	3.114
0.047	0.031	0.035	91.819	92.303	91.335	3.061
0.047	0.031	0.039	92.068	92.548	91.588	3.021
0.047	0.031	0.043	92.540	93.027	92.053	2.995
0.047	0.031	0.047	92.670	93.130	92.210	2.965
0.047	0.031	0.051	92.791	93.241	92.340	2.949
0.047	0.031	0.055	92.722	93.210	92.234	2.942
0.047	0.063	0.039	90.478	90.978	89.978	2.511
0.047	0.063	0.043	90.688	91.256	90.121	2.483
0.047	0.063	0.047	90.921	91.457	90.386	2.465
0.047	0.063	0.051	91.007	91.585	90.428	2.448
0.047	0.063	0.055	91.183	91.727	90.640	2.447

Continued on next page

Table 3. ViT-MAE results on CIFAR100.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.047	0.063	0.059	91.335	91.857	90.812	2.438
0.047	0.063	0.063	91.357	91.884	90.831	2.442
0.047	0.063	0.067	91.404	91.893	90.916	2.456
0.047	0.063	0.071	91.182	91.643	90.722	2.465
0.047	0.063	0.075	90.993	91.522	90.464	2.478
0.047	0.063	0.078	90.871	91.422	90.319	2.486
0.047	0.063	0.082	90.801	91.323	90.280	2.505
0.047	0.063	0.086	90.611	91.162	90.060	2.532
0.063	0.031	0.008	81.099	81.796	80.402	4.461
0.063	0.031	0.012	83.810	84.478	83.143	4.249
0.063	0.031	0.016	86.129	86.751	85.507	4.065
0.063	0.031	0.020	87.868	88.478	87.258	3.881
0.063	0.031	0.024	89.181	89.776	88.585	3.708
0.063	0.031	0.027	90.033	90.548	89.518	3.574
0.063	0.031	0.031	90.809	91.336	90.283	3.460
0.063	0.031	0.035	91.366	91.878	90.854	3.352
0.063	0.031	0.039	91.845	92.339	91.352	3.284
0.063	0.031	0.043	92.227	92.704	91.749	3.234
0.063	0.031	0.047	92.430	92.936	91.923	3.180
0.063	0.031	0.051	92.673	93.158	92.187	3.153
0.063	0.031	0.055	92.805	93.275	92.335	3.133
0.063	0.063	0.039	89.318	89.904	88.731	2.361
0.063	0.063	0.043	89.521	90.111	88.932	2.327
0.063	0.063	0.047	89.823	90.362	89.284	2.306
0.063	0.063	0.051	90.139	90.637	89.642	2.284
0.063	0.063	0.055	90.336	90.901	89.770	2.272
0.063	0.063	0.059	90.381	90.920	89.842	2.264
0.063	0.063	0.063	90.508	91.020	89.996	2.266
0.063	0.063	0.067	90.523	91.074	89.972	2.262
0.063	0.063	0.071	90.526	91.012	90.039	2.260
0.063	0.063	0.075	90.474	91.010	89.938	2.274
0.063	0.063	0.078	90.425	90.980	89.869	2.287
0.063	0.063	0.082	90.362	90.883	89.841	2.305
0.063	0.063	0.086	90.321	90.844	89.797	2.327

Table 4. ViT-MAE results on MNIST.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	91.758	92.259	91.256	2.815
0.000	0.031	0.012	91.548	92.039	91.056	2.821
0.000	0.031	0.016	91.279	91.805	90.752	2.825
0.000	0.031	0.020	91.125	91.590	90.659	2.827
0.000	0.031	0.024	91.004	91.535	90.472	2.831
0.000	0.031	0.027	90.726	91.317	90.135	2.833
0.000	0.031	0.031	90.460	90.998	89.922	2.836
0.000	0.031	0.035	90.111	90.660	89.563	2.837
0.000	0.031	0.039	89.821	90.319	89.323	2.845
0.000	0.031	0.043	89.687	90.232	89.142	2.848
0.000	0.031	0.047	89.502	90.124	88.879	2.856
0.000	0.031	0.051	89.214	89.784	88.644	2.859
0.000	0.031	0.055	88.869	89.428	88.310	2.865
0.000	0.063	0.039	91.807	92.297	91.317	3.099
0.000	0.063	0.043	91.482	92.006	90.958	3.100
0.000	0.063	0.047	91.285	91.799	90.771	3.103
0.000	0.063	0.051	91.030	91.528	90.532	3.113
0.000	0.063	0.055	90.791	91.318	90.264	3.115
0.000	0.063	0.059	90.587	91.103	90.072	3.121

Continued on next page

Table 4. ViT-MAE results on MNIST.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.063	0.063	90.328	90.805	89.851	3.129
0.000	0.063	0.067	90.119	90.693	89.545	3.131
0.000	0.063	0.071	89.920	90.486	89.353	3.134
0.000	0.063	0.075	89.744	90.289	89.199	3.138
0.000	0.063	0.078	89.424	89.977	88.870	3.142
0.000	0.063	0.082	89.256	89.846	88.666	3.149
0.000	0.063	0.086	89.045	89.636	88.454	3.155
0.031	0.031	0.008	91.861	92.343	91.380	0.921
0.031	0.031	0.012	91.630	92.170	91.090	0.919
0.031	0.031	0.016	91.179	91.713	90.645	0.916
0.031	0.031	0.020	90.729	91.268	90.189	0.912
0.031	0.031	0.024	90.312	90.835	89.788	0.908
0.031	0.031	0.027	89.652	90.218	89.086	0.901
0.031	0.031	0.031	89.290	89.856	88.724	0.898
0.031	0.031	0.035	88.811	89.341	88.281	0.893
0.031	0.031	0.039	88.578	89.150	88.006	0.891
0.031	0.031	0.043	87.905	88.419	87.392	0.885
0.031	0.031	0.047	87.504	88.112	86.895	0.881
0.031	0.031	0.051	87.091	87.744	86.438	0.877
0.031	0.031	0.055	86.436	87.113	85.760	0.872
0.031	0.063	0.039	91.703	92.220	91.186	0.928
0.031	0.063	0.043	91.350	91.901	90.800	0.925
0.031	0.063	0.047	90.874	91.371	90.377	0.921
0.031	0.063	0.051	90.565	91.071	90.058	0.918
0.031	0.063	0.055	90.020	90.536	89.504	0.914
0.031	0.063	0.059	89.606	90.181	89.031	0.910
0.031	0.063	0.063	89.165	89.738	88.592	0.907
0.031	0.063	0.067	88.764	89.389	88.139	0.904
0.031	0.063	0.071	88.332	88.951	87.714	0.900
0.031	0.063	0.075	87.875	88.424	87.326	0.897
0.031	0.063	0.078	87.538	88.087	86.989	0.894
0.031	0.063	0.082	87.158	87.755	86.560	0.892
0.031	0.063	0.086	86.618	87.318	85.919	0.888
0.047	0.031	0.008	91.857	92.308	91.407	0.926
0.047	0.031	0.012	91.653	92.123	91.183	0.924
0.047	0.031	0.016	91.340	91.891	90.789	0.923
0.047	0.031	0.020	91.066	91.582	90.550	0.920
0.047	0.031	0.024	90.798	91.319	90.277	0.918
0.047	0.031	0.027	90.603	91.143	90.064	0.916
0.047	0.031	0.031	90.313	90.840	89.787	0.914
0.047	0.031	0.035	90.016	90.561	89.470	0.912
0.047	0.031	0.039	89.549	90.105	88.994	0.908
0.047	0.031	0.043	89.226	89.773	88.679	0.906
0.047	0.031	0.047	88.776	89.384	88.169	0.902
0.047	0.031	0.051	88.467	89.040	87.893	0.899
0.047	0.031	0.055	88.129	88.753	87.504	0.896
0.047	0.063	0.039	91.643	92.155	91.130	0.932
0.047	0.063	0.043	91.388	91.926	90.850	0.930
0.047	0.063	0.047	91.078	91.599	90.557	0.928
0.047	0.063	0.051	90.862	91.378	90.346	0.927
0.047	0.063	0.055	90.754	91.267	90.241	0.926
0.047	0.063	0.059	90.432	90.982	89.882	0.924
0.047	0.063	0.063	89.963	90.466	89.460	0.921
0.047	0.063	0.067	89.522	90.064	88.980	0.917
0.047	0.063	0.071	89.284	89.828	88.740	0.915
0.047	0.063	0.075	88.837	89.400	88.274	0.911
0.047	0.063	0.078	88.606	89.184	88.028	0.910
0.047	0.063	0.082	88.149	88.726	87.572	0.907
0.047	0.063	0.086	87.740	88.374	87.107	0.904

Continued on next page

Table 4. ViT-MAE results on MNIST.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.063	0.031	0.008	90.583	91.083	90.083	0.913
0.063	0.031	0.012	90.392	90.976	89.807	0.911
0.063	0.031	0.016	90.103	90.687	89.518	0.908
0.063	0.031	0.020	89.824	90.363	89.285	0.906
0.063	0.031	0.024	89.573	90.116	89.030	0.903
0.063	0.031	0.027	89.317	89.862	88.773	0.901
0.063	0.031	0.031	89.149	89.729	88.569	0.900
0.063	0.031	0.035	89.039	89.569	88.509	0.899
0.063	0.031	0.039	88.605	89.215	87.996	0.895
0.063	0.031	0.043	88.377	88.989	87.765	0.893
0.063	0.031	0.047	88.144	88.746	87.543	0.891
0.063	0.031	0.051	87.920	88.466	87.374	0.889
0.063	0.031	0.055	87.660	88.271	87.050	0.887
0.063	0.063	0.039	90.405	90.942	89.867	0.915
0.063	0.063	0.043	90.241	90.775	89.707	0.914
0.063	0.063	0.047	89.953	90.500	89.406	0.912
0.063	0.063	0.051	89.806	90.326	89.285	0.911
0.063	0.063	0.055	89.658	90.236	89.080	0.909
0.063	0.063	0.059	89.455	90.013	88.896	0.908
0.063	0.063	0.063	89.068	89.642	88.494	0.904
0.063	0.063	0.067	88.843	89.392	88.294	0.903
0.063	0.063	0.071	88.411	89.051	87.772	0.899
0.063	0.063	0.075	88.108	88.689	87.527	0.896
0.063	0.063	0.078	87.968	88.538	87.397	0.895
0.063	0.063	0.082	87.898	88.555	87.242	0.895
0.063	0.063	0.086	87.741	88.346	87.135	0.894

Table 5. ViT-MAE results on TINYIMAGENET.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	96.487	96.820	96.154	170.019
0.000	0.031	0.012	95.553	95.885	95.221	168.337
0.000	0.031	0.016	94.815	95.233	94.397	166.617
0.000	0.031	0.020	93.665	94.088	93.243	164.929
0.000	0.031	0.024	92.283	92.804	91.761	163.257
0.000	0.031	0.027	91.099	91.625	90.573	161.673
0.000	0.031	0.031	89.778	90.325	89.230	160.208
0.000	0.031	0.035	88.793	89.373	88.213	158.684
0.000	0.031	0.039	87.607	88.251	86.963	157.348
0.000	0.031	0.043	86.556	87.128	85.984	156.070
0.000	0.031	0.047	85.678	86.351	85.005	154.842
0.000	0.031	0.051	84.973	85.641	84.305	153.738
0.000	0.031	0.055	84.097	84.785	83.410	152.661
0.000	0.063	0.039	93.520	93.978	93.063	178.312
0.000	0.063	0.043	92.847	93.311	92.382	177.318
0.000	0.063	0.047	92.154	92.665	91.643	176.360
0.000	0.063	0.051	91.371	91.900	90.841	175.419
0.000	0.063	0.055	90.713	91.220	90.205	174.613
0.000	0.063	0.059	90.198	90.695	89.702	173.806
0.000	0.063	0.063	89.704	90.241	89.168	173.052
0.000	0.063	0.067	89.132	89.700	88.565	172.400
0.000	0.063	0.071	88.632	89.175	88.089	171.706
0.000	0.063	0.075	88.374	88.986	87.761	171.218
0.000	0.063	0.078	88.034	88.627	87.441	170.699
0.000	0.063	0.082	87.621	88.216	87.026	170.192
0.000	0.063	0.086	87.259	87.841	86.678	169.761
0.016	0.031	0.008	89.869	90.398	89.340	141.321

Continued on next page

Table 5. ViT-MAE results on TINYIMAGENET.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.016	0.031	0.012	91.954	92.432	91.476	137.997
0.016	0.031	0.016	92.647	93.099	92.195	136.134
0.016	0.031	0.020	92.538	92.972	92.105	135.497
0.016	0.031	0.024	92.211	92.699	91.723	135.668
0.016	0.031	0.027	91.514	92.021	91.006	136.317
0.016	0.031	0.031	90.246	90.845	89.646	137.348
0.016	0.031	0.035	88.771	89.359	88.183	138.477
0.016	0.031	0.039	87.120	87.684	86.557	139.753
0.016	0.031	0.043	85.195	85.823	84.566	141.049
0.016	0.031	0.047	83.318	83.932	82.704	142.306
0.016	0.031	0.051	81.128	81.841	80.415	143.523
0.016	0.031	0.055	79.358	80.078	78.637	144.672
0.016	0.063	0.039	95.556	95.930	95.182	169.680
0.016	0.063	0.043	94.854	95.234	94.475	170.582
0.016	0.063	0.047	93.862	94.274	93.450	171.465
0.016	0.063	0.051	92.611	93.125	92.096	172.307
0.016	0.063	0.055	91.523	92.000	91.045	173.135
0.016	0.063	0.059	90.581	91.126	90.036	173.919
0.016	0.063	0.063	89.544	90.083	89.005	174.630
0.016	0.063	0.067	88.618	89.239	87.997	175.363
0.016	0.063	0.071	87.585	88.263	86.907	176.051
0.016	0.063	0.075	86.516	87.089	85.943	176.622
0.016	0.063	0.078	85.766	86.388	85.143	177.224
0.016	0.063	0.082	85.025	85.708	84.342	177.798
0.016	0.063	0.086	84.049	84.720	83.378	178.344
0.031	0.031	0.008	90.938	91.430	90.445	176.989
0.031	0.031	0.012	91.271	91.796	90.746	177.540
0.031	0.031	0.016	91.419	91.936	90.902	177.920
0.031	0.031	0.020	91.517	92.006	91.027	178.009
0.031	0.031	0.024	91.430	91.962	90.898	177.923
0.031	0.031	0.027	91.350	91.846	90.853	177.823
0.031	0.031	0.031	91.197	91.719	90.676	177.637
0.031	0.031	0.035	90.955	91.509	90.401	177.362
0.031	0.031	0.039	90.380	90.935	89.825	177.044
0.031	0.031	0.043	89.888	90.449	89.328	176.603
0.031	0.031	0.047	89.476	90.064	88.888	176.164
0.031	0.031	0.051	88.991	89.595	88.387	175.774
0.031	0.031	0.055	88.372	88.915	87.829	175.245
0.031	0.063	0.039	93.441	93.876	93.007	183.661
0.031	0.063	0.043	93.083	93.567	92.599	183.370
0.031	0.063	0.047	92.759	93.224	92.293	183.060
0.031	0.063	0.051	92.069	92.544	91.594	182.647
0.031	0.063	0.055	91.694	92.196	91.192	182.258
0.031	0.063	0.059	91.303	91.797	90.809	181.835
0.031	0.063	0.063	90.875	91.394	90.357	181.421
0.031	0.063	0.067	90.375	90.884	89.865	180.932
0.031	0.063	0.071	89.911	90.443	89.378	180.445
0.031	0.063	0.075	89.514	90.039	88.989	179.971
0.031	0.063	0.078	88.958	89.529	88.387	179.477
0.031	0.063	0.082	88.399	88.967	87.830	178.984
0.031	0.063	0.086	87.756	88.333	87.180	178.432
0.047	0.031	0.008	84.027	84.681	83.372	133.348
0.047	0.031	0.012	85.681	86.318	85.043	134.370
0.047	0.031	0.016	87.023	87.639	86.407	134.994
0.047	0.031	0.020	87.895	88.483	87.308	135.318
0.047	0.031	0.024	88.668	89.238	88.099	135.555
0.047	0.031	0.027	89.261	89.805	88.718	135.610
0.047	0.031	0.031	89.707	90.247	89.168	135.786
0.047	0.031	0.035	90.094	90.595	89.593	135.841

Continued on next page

Table 5. ViT-MAE results on TINYIMAGENET.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.047	0.031	0.039	90.165	90.706	89.624	135.939
0.047	0.031	0.043	90.428	90.891	89.965	135.900
0.047	0.031	0.047	90.738	91.286	90.190	135.950
0.047	0.031	0.051	90.821	91.400	90.242	136.025
0.047	0.031	0.055	90.842	91.368	90.316	136.130
0.047	0.063	0.039	89.242	89.820	88.663	130.539
0.047	0.063	0.043	89.428	90.014	88.843	130.609
0.047	0.063	0.047	89.775	90.336	89.214	130.669
0.047	0.063	0.051	89.885	90.377	89.394	130.708
0.047	0.063	0.055	89.858	90.389	89.327	130.757
0.047	0.063	0.059	90.066	90.637	89.494	130.869
0.047	0.063	0.063	90.048	90.623	89.473	130.932
0.047	0.063	0.067	90.038	90.584	89.491	131.031
0.047	0.063	0.071	90.065	90.615	89.515	131.078
0.047	0.063	0.075	90.217	90.750	89.685	131.127
0.047	0.063	0.078	90.132	90.705	89.560	131.240
0.047	0.063	0.082	90.101	90.642	89.561	131.246
0.047	0.063	0.086	90.094	90.617	89.572	131.359
0.063	0.031	0.008	75.885	76.626	75.144	128.690
0.063	0.031	0.012	80.583	81.340	79.826	130.491
0.063	0.031	0.016	83.702	84.367	83.036	131.670
0.063	0.031	0.020	86.077	86.633	85.522	132.468
0.063	0.031	0.024	87.898	88.467	87.328	132.912
0.063	0.031	0.027	89.421	89.995	88.846	133.285
0.063	0.031	0.031	90.456	91.028	89.884	133.472
0.063	0.031	0.035	91.231	91.748	90.713	133.715
0.063	0.031	0.039	91.909	92.391	91.428	133.866
0.063	0.031	0.043	92.550	93.006	92.095	133.956
0.063	0.031	0.047	93.235	93.683	92.787	134.037
0.063	0.031	0.051	93.601	94.054	93.147	134.119
0.063	0.031	0.055	93.878	94.313	93.444	134.139
0.063	0.063	0.039	87.402	88.024	86.780	112.297
0.063	0.063	0.043	88.207	88.806	87.607	112.443
0.063	0.063	0.047	88.727	89.293	88.161	112.526
0.063	0.063	0.051	89.292	89.873	88.710	112.579
0.063	0.063	0.055	89.520	90.065	88.976	112.685
0.063	0.063	0.059	89.971	90.506	89.436	112.707
0.063	0.063	0.063	90.154	90.688	89.619	112.784
0.063	0.063	0.067	90.232	90.771	89.693	112.797
0.063	0.063	0.071	90.430	90.956	89.904	112.821
0.063	0.063	0.075	90.299	90.850	89.748	112.845
0.063	0.063	0.078	90.361	90.886	89.836	112.925
0.063	0.063	0.082	90.340	90.911	89.768	113.037
0.063	0.063	0.086	90.267	90.810	89.723	113.122

Table 6. ResNet50D results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	94.391	94.786	93.997	6.925
0.000	0.031	0.012	93.544	93.997	93.091	6.949
0.000	0.031	0.016	92.637	93.104	92.171	6.958
0.000	0.031	0.020	91.939	92.418	91.459	6.956
0.000	0.031	0.024	91.068	91.615	90.520	6.957
0.000	0.031	0.027	90.240	90.779	89.702	6.954
0.000	0.031	0.031	89.463	90.079	88.847	6.951
0.000	0.031	0.035	88.792	89.439	88.145	6.946
0.000	0.031	0.039	88.142	88.778	87.505	6.951

Continued on next page

Table 6. ResNet50D results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.043	87.629	88.222	87.036	6.959
0.000	0.031	0.047	87.157	87.761	86.553	6.962
0.000	0.031	0.051	86.786	87.367	86.204	6.970
0.000	0.031	0.055	86.528	87.153	85.903	6.976
0.000	0.063	0.039	90.864	91.388	90.340	7.429
0.000	0.063	0.043	90.387	90.928	89.846	7.427
0.000	0.063	0.047	89.883	90.413	89.354	7.429
0.000	0.063	0.051	89.454	90.015	88.893	7.422
0.000	0.063	0.055	89.244	89.828	88.659	7.433
0.000	0.063	0.059	89.123	89.664	88.582	7.436
0.000	0.063	0.063	88.873	89.423	88.322	7.452
0.000	0.063	0.067	88.818	89.404	88.232	7.462
0.000	0.063	0.071	88.642	89.221	88.063	7.484
0.000	0.063	0.075	88.451	89.039	87.863	7.493
0.000	0.063	0.078	88.401	88.992	87.810	7.520
0.000	0.063	0.082	88.250	88.836	87.665	7.540
0.000	0.063	0.086	88.125	88.751	87.500	7.555
0.016	0.031	0.008	94.245	94.726	93.764	4.749
0.016	0.031	0.012	93.677	94.100	93.254	4.759
0.016	0.031	0.016	93.225	93.695	92.756	4.759
0.016	0.031	0.020	92.724	93.168	92.281	4.761
0.016	0.031	0.024	92.106	92.592	91.620	4.752
0.016	0.031	0.027	91.635	92.173	91.098	4.748
0.016	0.031	0.031	90.837	91.329	90.345	4.745
0.016	0.031	0.035	90.094	90.693	89.495	4.739
0.016	0.031	0.039	89.341	89.892	88.789	4.730
0.016	0.031	0.043	88.624	89.223	88.025	4.720
0.016	0.031	0.047	87.867	88.500	87.234	4.705
0.016	0.031	0.051	87.180	87.789	86.571	4.694
0.016	0.031	0.055	86.481	87.160	85.801	4.679
0.016	0.063	0.039	94.032	94.449	93.615	5.521
0.016	0.063	0.043	93.473	93.936	93.010	5.514
0.016	0.063	0.047	93.092	93.549	92.635	5.505
0.016	0.063	0.051	92.581	93.110	92.053	5.486
0.016	0.063	0.055	92.091	92.559	91.623	5.473
0.016	0.063	0.059	91.343	91.848	90.837	5.457
0.016	0.063	0.063	90.862	91.354	90.369	5.442
0.016	0.063	0.067	90.043	90.588	89.498	5.427
0.016	0.063	0.071	89.400	89.945	88.856	5.408
0.016	0.063	0.075	89.053	89.658	88.449	5.394
0.016	0.063	0.078	88.505	89.067	87.943	5.382
0.016	0.063	0.082	87.719	88.322	87.116	5.360
0.016	0.063	0.086	87.102	87.714	86.491	5.342
0.031	0.031	0.008	92.929	93.397	92.462	4.562
0.031	0.031	0.012	92.470	92.956	91.984	4.571
0.031	0.031	0.016	92.082	92.589	91.575	4.576
0.031	0.031	0.020	91.585	92.094	91.076	4.581
0.031	0.031	0.024	91.240	91.764	90.716	4.583
0.031	0.031	0.027	90.768	91.315	90.221	4.583
0.031	0.031	0.031	90.304	90.814	89.794	4.583
0.031	0.031	0.035	89.623	90.141	89.105	4.577
0.031	0.031	0.039	89.089	89.631	88.547	4.576
0.031	0.031	0.043	88.572	89.139	88.005	4.572
0.031	0.031	0.047	88.059	88.640	87.478	4.573
0.031	0.031	0.051	87.356	87.925	86.787	4.565
0.031	0.031	0.055	86.682	87.279	86.085	4.557
0.031	0.063	0.039	93.270	93.702	92.838	5.271
0.031	0.063	0.043	92.820	93.314	92.325	5.271
0.031	0.063	0.047	92.274	92.809	91.739	5.265

Continued on next page

Table 6. ResNet50D results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.031	0.063	0.051	91.710	92.229	91.191	5.253
0.031	0.063	0.055	91.381	91.882	90.880	5.254
0.031	0.063	0.059	90.946	91.473	90.418	5.249
0.031	0.063	0.063	90.496	91.040	89.952	5.248
0.031	0.063	0.067	89.943	90.570	89.316	5.245
0.031	0.063	0.071	89.517	90.089	88.944	5.233
0.031	0.063	0.075	89.110	89.656	88.564	5.225
0.031	0.063	0.078	88.465	89.047	87.883	5.211
0.031	0.063	0.082	87.955	88.585	87.325	5.205
0.031	0.063	0.086	87.539	88.158	86.919	5.198
0.047	0.031	0.008	91.953	92.442	91.464	4.559
0.047	0.031	0.012	91.530	92.033	91.027	4.563
0.047	0.031	0.016	91.340	91.855	90.826	4.571
0.047	0.031	0.020	91.032	91.562	90.503	4.573
0.047	0.031	0.024	90.771	91.334	90.207	4.578
0.047	0.031	0.027	90.452	91.013	89.892	4.581
0.047	0.031	0.031	90.150	90.662	89.638	4.582
0.047	0.031	0.035	89.994	90.542	89.446	4.588
0.047	0.031	0.039	89.638	90.227	89.050	4.590
0.047	0.031	0.043	89.331	89.905	88.757	4.591
0.047	0.031	0.047	88.918	89.458	88.379	4.589
0.047	0.031	0.051	88.567	89.128	88.006	4.591
0.047	0.031	0.055	88.227	88.770	87.683	4.591
0.047	0.063	0.039	91.796	92.298	91.295	5.034
0.047	0.063	0.043	91.653	92.164	91.141	5.037
0.047	0.063	0.047	91.337	91.867	90.807	5.041
0.047	0.063	0.051	90.937	91.451	90.424	5.040
0.047	0.063	0.055	90.624	91.128	90.120	5.035
0.047	0.063	0.059	90.402	90.937	89.867	5.042
0.047	0.063	0.063	90.122	90.639	89.605	5.038
0.047	0.063	0.067	89.754	90.292	89.216	5.042
0.047	0.063	0.071	89.492	90.034	88.951	5.041
0.047	0.063	0.075	89.226	89.812	88.640	5.041
0.047	0.063	0.078	88.979	89.509	88.448	5.040
0.047	0.063	0.082	88.659	89.213	88.106	5.040
0.047	0.063	0.086	88.394	89.008	87.780	5.037
0.063	0.031	0.008	91.742	92.270	91.214	4.732
0.063	0.031	0.012	91.493	92.025	90.962	4.734
0.063	0.031	0.016	91.280	91.784	90.776	4.736
0.063	0.031	0.020	90.985	91.499	90.471	4.742
0.063	0.031	0.024	90.677	91.187	90.168	4.742
0.063	0.031	0.027	90.352	90.914	89.790	4.741
0.063	0.031	0.031	89.991	90.489	89.492	4.748
0.063	0.031	0.035	89.818	90.323	89.314	4.751
0.063	0.031	0.039	89.549	90.089	89.009	4.754
0.063	0.031	0.043	89.214	89.787	88.642	4.750
0.063	0.031	0.047	88.897	89.494	88.300	4.750
0.063	0.031	0.051	88.635	89.202	88.067	4.751
0.063	0.031	0.055	88.362	88.923	87.801	4.751
0.063	0.063	0.039	91.901	92.398	91.405	5.150
0.063	0.063	0.043	91.639	92.156	91.122	5.150
0.063	0.063	0.047	91.358	91.834	90.882	5.152
0.063	0.063	0.051	91.150	91.661	90.639	5.152
0.063	0.063	0.055	90.964	91.496	90.432	5.152
0.063	0.063	0.059	90.767	91.290	90.244	5.154
0.063	0.063	0.063	90.466	91.001	89.932	5.154
0.063	0.063	0.067	90.172	90.714	89.630	5.151
0.063	0.063	0.071	90.001	90.516	89.486	5.154
0.063	0.063	0.075	89.765	90.330	89.200	5.155

Continued on next page

Table 6. ResNet50D results on CIFAR10.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.063	0.063	0.078	89.598	90.117	89.079	5.155
0.063	0.063	0.082	89.237	89.870	88.604	5.151
0.063	0.063	0.086	88.917	89.478	88.357	5.154

Table 7. ResNet50D results on CIFAR100.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	96.849	97.154	96.544	82.209
0.000	0.031	0.012	96.043	96.429	95.656	82.345
0.000	0.031	0.016	95.355	95.728	94.982	82.130
0.000	0.031	0.020	94.478	94.930	94.026	81.513
0.000	0.031	0.024	93.150	93.644	92.656	80.641
0.000	0.031	0.027	91.909	92.432	91.387	79.523
0.000	0.031	0.031	90.734	91.275	90.193	78.305
0.000	0.031	0.035	88.915	89.498	88.332	77.043
0.000	0.031	0.039	87.355	87.975	86.734	75.634
0.000	0.031	0.043	85.779	86.378	85.179	74.338
0.000	0.031	0.047	84.414	85.048	83.781	72.991
0.000	0.031	0.051	82.679	83.406	81.952	71.747
0.000	0.031	0.055	81.172	81.841	80.504	70.541
0.000	0.063	0.039	95.808	96.177	95.439	89.763
0.000	0.063	0.043	94.916	95.280	94.553	88.857
0.000	0.063	0.047	93.906	94.339	93.472	87.886
0.000	0.063	0.051	92.854	93.353	92.356	86.991
0.000	0.063	0.055	91.968	92.475	91.461	86.038
0.000	0.063	0.059	91.290	91.780	90.801	85.189
0.000	0.063	0.063	90.636	91.152	90.121	84.380
0.000	0.063	0.067	89.668	90.244	89.092	83.500
0.000	0.063	0.071	88.807	89.384	88.229	82.737
0.000	0.063	0.075	87.988	88.550	87.426	81.933
0.000	0.063	0.078	86.936	87.554	86.318	81.218
0.000	0.063	0.082	85.933	86.530	85.336	80.447
0.000	0.063	0.086	85.253	85.881	84.624	79.798
0.016	0.031	0.008	93.965	94.393	93.537	52.920
0.016	0.031	0.012	93.595	94.048	93.142	52.924
0.016	0.031	0.016	93.325	93.756	92.895	52.941
0.016	0.031	0.020	93.105	93.537	92.673	52.948
0.016	0.031	0.024	92.504	93.036	91.973	52.957
0.016	0.031	0.027	91.935	92.449	91.421	52.936
0.016	0.031	0.031	91.397	91.920	90.873	52.896
0.016	0.031	0.035	90.973	91.488	90.458	52.851
0.016	0.031	0.039	90.609	91.152	90.066	52.803
0.016	0.031	0.043	90.050	90.638	89.461	52.766
0.016	0.031	0.047	89.505	90.060	88.949	52.704
0.016	0.031	0.051	88.650	89.194	88.106	52.650
0.016	0.031	0.055	88.070	88.655	87.484	52.579
0.016	0.063	0.039	93.694	94.123	93.265	61.594
0.016	0.063	0.043	93.316	93.783	92.848	61.549
0.016	0.063	0.047	93.186	93.638	92.735	61.476
0.016	0.063	0.051	92.907	93.415	92.399	61.431
0.016	0.063	0.055	92.479	92.958	92.000	61.372
0.016	0.063	0.059	92.179	92.626	91.732	61.293
0.016	0.063	0.063	91.603	92.140	91.067	61.198
0.016	0.063	0.067	90.978	91.443	90.512	61.120
0.016	0.063	0.071	90.399	90.920	89.878	61.017
0.016	0.063	0.075	90.006	90.565	89.447	60.921
0.016	0.063	0.078	89.563	90.098	89.027	60.811

Continued on next page

Table 7. ResNet50D results on CIFAR100.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.016	0.063	0.082	89.142	89.704	88.579	60.717
0.016	0.063	0.086	88.627	89.200	88.055	60.615
0.031	0.031	0.008	93.766	94.217	93.316	49.186
0.031	0.031	0.012	93.191	93.678	92.703	49.170
0.031	0.031	0.016	92.791	93.272	92.310	49.209
0.031	0.031	0.020	92.424	92.909	91.939	49.187
0.031	0.031	0.024	91.981	92.499	91.462	49.165
0.031	0.031	0.027	91.482	91.953	91.012	49.150
0.031	0.031	0.031	90.956	91.481	90.432	49.120
0.031	0.031	0.035	90.560	91.120	90.001	49.105
0.031	0.031	0.039	90.083	90.639	89.527	49.062
0.031	0.031	0.043	89.481	90.020	88.941	49.027
0.031	0.031	0.047	88.995	89.544	88.445	48.975
0.031	0.031	0.051	88.494	89.084	87.904	48.953
0.031	0.031	0.055	88.169	88.744	87.594	48.880
0.031	0.063	0.039	93.436	93.863	93.008	56.774
0.031	0.063	0.043	93.229	93.651	92.806	56.741
0.031	0.063	0.047	92.812	93.288	92.337	56.729
0.031	0.063	0.051	92.326	92.862	91.789	56.704
0.031	0.063	0.055	92.050	92.534	91.567	56.645
0.031	0.063	0.059	91.655	92.153	91.158	56.564
0.031	0.063	0.063	91.253	91.750	90.757	56.541
0.031	0.063	0.067	90.902	91.468	90.335	56.474
0.031	0.063	0.071	90.424	90.969	89.879	56.409
0.031	0.063	0.075	90.004	90.589	89.418	56.375
0.031	0.063	0.078	89.714	90.337	89.090	56.308
0.031	0.063	0.082	89.320	89.848	88.791	56.265
0.031	0.063	0.086	88.845	89.408	88.282	56.208
0.047	0.031	0.008	91.689	92.153	91.224	50.040
0.047	0.031	0.012	91.460	91.927	90.993	50.025
0.047	0.031	0.016	91.033	91.535	90.531	49.993
0.047	0.031	0.020	90.711	91.226	90.195	49.997
0.047	0.031	0.024	90.315	90.874	89.756	49.977
0.047	0.031	0.027	89.987	90.532	89.442	49.933
0.047	0.031	0.031	89.495	90.024	88.966	49.907
0.047	0.031	0.035	89.113	89.637	88.588	49.898
0.047	0.031	0.039	88.772	89.356	88.187	49.859
0.047	0.031	0.043	88.401	89.030	87.772	49.813
0.047	0.031	0.047	88.012	88.592	87.433	49.785
0.047	0.031	0.051	87.496	88.093	86.899	49.735
0.047	0.031	0.055	87.018	87.628	86.409	49.714
0.047	0.063	0.039	91.617	92.093	91.141	55.600
0.047	0.063	0.043	91.279	91.818	90.741	55.573
0.047	0.063	0.047	90.854	91.396	90.311	55.516
0.047	0.063	0.051	90.343	90.874	89.811	55.490
0.047	0.063	0.055	90.173	90.724	89.622	55.448
0.047	0.063	0.059	89.986	90.561	89.411	55.430
0.047	0.063	0.063	89.758	90.340	89.176	55.356
0.047	0.063	0.067	89.386	89.943	88.829	55.308
0.047	0.063	0.071	88.992	89.567	88.416	55.291
0.047	0.063	0.075	88.734	89.322	88.146	55.253
0.047	0.063	0.078	88.308	88.935	87.681	55.184
0.047	0.063	0.082	88.001	88.612	87.390	55.143
0.047	0.063	0.086	87.421	88.055	86.786	55.094
0.063	0.031	0.008	92.693	93.143	92.243	50.434
0.063	0.031	0.012	92.446	92.909	91.983	50.429
0.063	0.031	0.016	92.191	92.642	91.739	50.453
0.063	0.031	0.020	91.865	92.374	91.356	50.425
0.063	0.031	0.024	91.534	92.028	91.041	50.411

Continued on next page

Table 7. ResNet50D results on CIFAR100.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.063	0.031	0.027	91.176	91.658	90.695	50.389
0.063	0.031	0.031	90.644	91.196	90.093	50.357
0.063	0.031	0.035	90.368	90.888	89.848	50.360
0.063	0.031	0.039	89.736	90.258	89.213	50.337
0.063	0.031	0.043	89.343	89.852	88.834	50.330
0.063	0.031	0.047	89.191	89.763	88.619	50.293
0.063	0.031	0.051	88.888	89.425	88.350	50.254
0.063	0.031	0.055	88.602	89.146	88.058	50.237
0.063	0.063	0.039	92.393	92.879	91.908	55.056
0.063	0.063	0.043	92.177	92.633	91.721	55.030
0.063	0.063	0.047	91.800	92.259	91.340	54.995
0.063	0.063	0.051	91.339	91.830	90.848	54.974
0.063	0.063	0.055	90.927	91.444	90.410	54.908
0.063	0.063	0.059	90.718	91.255	90.182	54.924
0.063	0.063	0.063	90.352	90.874	89.831	54.880
0.063	0.063	0.067	90.031	90.572	89.490	54.841
0.063	0.063	0.071	89.719	90.265	89.174	54.806
0.063	0.063	0.075	89.510	90.044	88.976	54.768
0.063	0.063	0.078	89.149	89.661	88.638	54.727
0.063	0.063	0.082	88.812	89.373	88.251	54.671
0.063	0.063	0.086	88.462	89.074	87.850	54.611

Table 8. ResNet50D results on MNIST.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	89.822	90.407	89.237	8.331
0.000	0.031	0.012	89.842	90.405	89.278	8.331
0.000	0.031	0.016	89.819	90.364	89.275	8.331
0.000	0.031	0.020	89.782	90.351	89.213	8.331
0.000	0.031	0.024	89.763	90.279	89.246	8.331
0.000	0.031	0.027	89.833	90.367	89.298	8.332
0.000	0.031	0.031	89.791	90.333	89.249	8.332
0.000	0.031	0.035	89.761	90.320	89.203	8.332
0.000	0.031	0.039	89.766	90.338	89.194	8.331
0.000	0.031	0.043	89.737	90.253	89.222	8.332
0.000	0.031	0.047	89.708	90.286	89.130	8.332
0.000	0.031	0.051	89.666	90.231	89.101	8.332
0.000	0.031	0.055	89.735	90.288	89.181	8.331
0.000	0.063	0.039	89.859	90.422	89.295	8.339
0.000	0.063	0.043	89.852	90.424	89.281	8.339
0.000	0.063	0.047	89.827	90.351	89.303	8.339
0.000	0.063	0.051	89.775	90.308	89.243	8.338
0.000	0.063	0.055	89.744	90.259	89.229	8.337
0.000	0.063	0.059	89.710	90.242	89.179	8.338
0.000	0.063	0.063	89.748	90.281	89.215	8.337
0.000	0.063	0.067	89.762	90.246	89.277	8.337
0.000	0.063	0.071	89.719	90.232	89.205	8.336
0.000	0.063	0.075	89.689	90.193	89.186	8.336
0.000	0.063	0.078	89.662	90.196	89.127	8.337
0.000	0.063	0.082	89.661	90.206	89.117	8.336
0.000	0.063	0.086	89.653	90.206	89.100	8.336
0.016	0.031	0.008	91.168	91.706	90.630	0.913
0.016	0.031	0.012	90.917	91.419	90.414	0.910
0.016	0.031	0.016	90.809	91.305	90.313	0.909
0.016	0.031	0.020	90.566	91.071	90.060	0.907
0.016	0.031	0.024	90.470	91.010	89.930	0.906
0.016	0.031	0.027	90.359	90.866	89.852	0.905

Continued on next page

Table 8. ResNet50D results on MNIST.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.016	0.031	0.031	90.207	90.759	89.654	0.903
0.016	0.031	0.035	90.000	90.567	89.433	0.901
0.016	0.031	0.039	89.944	90.484	89.404	0.901
0.016	0.031	0.043	89.727	90.254	89.199	0.898
0.016	0.031	0.047	89.556	90.106	89.006	0.897
0.016	0.031	0.051	89.479	90.001	88.956	0.896
0.016	0.031	0.055	89.276	89.823	88.729	0.894
0.016	0.063	0.039	91.119	91.702	90.537	0.913
0.016	0.063	0.043	90.963	91.509	90.417	0.911
0.016	0.063	0.047	90.854	91.391	90.317	0.910
0.016	0.063	0.051	90.673	91.216	90.131	0.909
0.016	0.063	0.055	90.462	91.000	89.925	0.907
0.016	0.063	0.059	90.244	90.767	89.720	0.904
0.016	0.063	0.063	90.107	90.653	89.560	0.903
0.016	0.063	0.067	90.085	90.662	89.507	0.903
0.016	0.063	0.071	89.815	90.374	89.255	0.900
0.016	0.063	0.075	89.659	90.194	89.125	0.899
0.016	0.063	0.078	89.564	90.130	88.998	0.898
0.016	0.063	0.082	89.451	90.017	88.885	0.897
0.016	0.063	0.086	89.177	89.748	88.606	0.894
0.031	0.031	0.008	90.794	91.294	90.295	0.908
0.031	0.031	0.012	90.666	91.196	90.136	0.907
0.031	0.031	0.016	90.620	91.145	90.096	0.907
0.031	0.031	0.020	90.469	91.040	89.898	0.905
0.031	0.031	0.024	90.434	91.026	89.842	0.905
0.031	0.031	0.027	90.182	90.745	89.619	0.902
0.031	0.031	0.031	90.081	90.597	89.566	0.901
0.031	0.031	0.035	89.874	90.366	89.381	0.899
0.031	0.031	0.039	89.678	90.269	89.088	0.897
0.031	0.031	0.043	89.542	90.097	88.987	0.896
0.031	0.031	0.047	89.291	89.877	88.704	0.894
0.031	0.031	0.051	89.080	89.623	88.536	0.892
0.031	0.031	0.055	88.943	89.501	88.386	0.890
0.031	0.063	0.039	90.915	91.458	90.372	0.910
0.031	0.063	0.043	90.774	91.274	90.273	0.909
0.031	0.063	0.047	90.576	91.078	90.075	0.907
0.031	0.063	0.051	90.501	90.976	90.026	0.906
0.031	0.063	0.055	90.370	90.927	89.813	0.905
0.031	0.063	0.059	90.257	90.725	89.789	0.904
0.031	0.063	0.063	90.094	90.633	89.555	0.902
0.031	0.063	0.067	89.948	90.446	89.451	0.901
0.031	0.063	0.071	89.751	90.299	89.203	0.899
0.031	0.063	0.075	89.536	90.117	88.955	0.897
0.031	0.063	0.078	89.485	90.040	88.930	0.896
0.031	0.063	0.082	89.323	89.899	88.748	0.895
0.031	0.063	0.086	89.220	89.754	88.686	0.894
0.047	0.031	0.008	92.184	92.679	91.688	0.924
0.047	0.031	0.012	92.092	92.587	91.598	0.923
0.047	0.031	0.016	91.842	92.352	91.332	0.921
0.047	0.031	0.020	91.807	92.260	91.355	0.921
0.047	0.031	0.024	91.785	92.293	91.278	0.921
0.047	0.031	0.027	91.646	92.159	91.134	0.919
0.047	0.031	0.031	91.482	91.995	90.968	0.918
0.047	0.031	0.035	91.289	91.787	90.791	0.916
0.047	0.031	0.039	91.205	91.743	90.667	0.915
0.047	0.031	0.043	90.999	91.529	90.470	0.913
0.047	0.031	0.047	90.875	91.459	90.291	0.912
0.047	0.031	0.051	90.745	91.298	90.192	0.911
0.047	0.031	0.055	90.641	91.201	90.080	0.910

Continued on next page

Table 8. ResNet50D results on MNIST.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.047	0.063	0.039	92.221	92.743	91.699	0.926
0.047	0.063	0.043	92.113	92.566	91.659	0.925
0.047	0.063	0.047	92.009	92.440	91.578	0.924
0.047	0.063	0.051	91.861	92.301	91.421	0.922
0.047	0.063	0.055	91.779	92.226	91.333	0.922
0.047	0.063	0.059	91.588	92.107	91.070	0.920
0.047	0.063	0.063	91.478	91.951	91.005	0.919
0.047	0.063	0.067	91.270	91.811	90.728	0.917
0.047	0.063	0.071	91.175	91.706	90.644	0.916
0.047	0.063	0.075	91.076	91.615	90.538	0.915
0.047	0.063	0.078	90.806	91.308	90.304	0.912
0.047	0.063	0.082	90.772	91.281	90.263	0.912
0.047	0.063	0.086	90.689	91.198	90.179	0.911
0.063	0.031	0.008	90.371	90.905	89.838	0.904
0.063	0.031	0.012	90.225	90.787	89.662	0.902
0.063	0.031	0.016	90.193	90.725	89.661	0.902
0.063	0.031	0.020	89.943	90.479	89.407	0.900
0.063	0.031	0.024	89.770	90.285	89.256	0.898
0.063	0.031	0.027	89.506	90.072	88.939	0.896
0.063	0.031	0.031	89.463	90.025	88.900	0.895
0.063	0.031	0.035	89.404	90.010	88.799	0.895
0.063	0.031	0.039	89.253	89.829	88.676	0.893
0.063	0.031	0.043	89.041	89.605	88.476	0.891
0.063	0.031	0.047	88.950	89.495	88.405	0.890
0.063	0.031	0.051	88.846	89.456	88.236	0.889
0.063	0.031	0.055	88.559	89.122	87.996	0.886
0.063	0.063	0.039	90.405	90.964	89.846	0.905
0.063	0.063	0.043	90.217	90.704	89.731	0.903
0.063	0.063	0.047	90.087	90.625	89.550	0.902
0.063	0.063	0.051	89.884	90.397	89.370	0.900
0.063	0.063	0.055	89.734	90.296	89.172	0.898
0.063	0.063	0.059	89.644	90.133	89.154	0.897
0.063	0.063	0.063	89.450	90.030	88.871	0.895
0.063	0.063	0.067	89.339	89.862	88.817	0.894
0.063	0.063	0.071	89.277	89.837	88.716	0.894
0.063	0.063	0.075	89.224	89.759	88.689	0.893
0.063	0.063	0.078	89.016	89.592	88.441	0.891
0.063	0.063	0.082	88.864	89.385	88.344	0.890
0.063	0.063	0.086	88.635	89.218	88.052	0.888

Table 9. ResNet50D results on TINYIMAGENET.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.031	0.008	94.616	95.061	94.170	173.363
0.000	0.031	0.012	93.821	94.262	93.379	171.958
0.000	0.031	0.016	93.158	93.614	92.702	170.702
0.000	0.031	0.020	92.467	92.935	91.998	169.504
0.000	0.031	0.024	91.806	92.328	91.284	168.403
0.000	0.031	0.027	91.072	91.636	90.509	167.274
0.000	0.031	0.031	90.510	91.050	89.971	166.433
0.000	0.031	0.035	89.909	90.420	89.398	165.565
0.000	0.031	0.039	89.161	89.743	88.579	164.719
0.000	0.031	0.043	88.611	89.204	88.017	164.049
0.000	0.031	0.047	88.017	88.603	87.431	163.473
0.000	0.031	0.051	87.662	88.332	86.992	162.969
0.000	0.031	0.055	87.310	87.886	86.734	162.525
0.000	0.063	0.039	92.756	93.234	92.279	175.310

Continued on next page

Table 9. ResNet50D results on TINYIMAGENET.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.000	0.063	0.043	92.301	92.819	91.783	174.690
0.000	0.063	0.047	92.046	92.536	91.556	174.134
0.000	0.063	0.051	91.695	92.198	91.192	173.674
0.000	0.063	0.055	91.245	91.801	90.690	173.275
0.000	0.063	0.059	90.901	91.441	90.362	172.935
0.000	0.063	0.063	90.707	91.277	90.136	172.609
0.000	0.063	0.067	90.450	90.975	89.926	172.338
0.000	0.063	0.071	90.223	90.745	89.701	172.165
0.000	0.063	0.075	89.981	90.562	89.400	171.849
0.000	0.063	0.078	89.813	90.358	89.268	171.753
0.000	0.063	0.082	89.559	90.122	88.996	171.609
0.000	0.063	0.086	89.439	90.011	88.868	171.540
0.016	0.031	0.008	92.834	93.323	92.345	181.635
0.016	0.031	0.012	92.355	92.829	91.880	181.238
0.016	0.031	0.016	91.955	92.490	91.420	180.822
0.016	0.031	0.020	91.321	91.804	90.838	180.369
0.016	0.031	0.024	90.706	91.239	90.173	179.941
0.016	0.031	0.027	90.214	90.734	89.695	179.532
0.016	0.031	0.031	89.653	90.209	89.098	179.144
0.016	0.031	0.035	88.885	89.456	88.315	178.652
0.016	0.031	0.039	88.355	88.998	87.712	178.257
0.016	0.031	0.043	87.555	88.216	86.895	177.771
0.016	0.031	0.047	86.955	87.548	86.362	177.345
0.016	0.031	0.051	86.299	86.871	85.727	176.898
0.016	0.031	0.055	85.653	86.291	85.016	176.417
0.016	0.063	0.039	92.456	92.923	91.989	186.094
0.016	0.063	0.043	91.900	92.400	91.400	185.759
0.016	0.063	0.047	91.319	91.837	90.801	185.438
0.016	0.063	0.051	90.862	91.405	90.319	185.105
0.016	0.063	0.055	90.507	91.047	89.967	184.754
0.016	0.063	0.059	89.914	90.496	89.332	184.387
0.016	0.063	0.063	89.498	90.055	88.941	184.073
0.016	0.063	0.067	89.045	89.591	88.499	183.706
0.016	0.063	0.071	88.662	89.219	88.105	183.364
0.016	0.063	0.075	88.176	88.775	87.578	183.010
0.016	0.063	0.078	87.533	88.191	86.875	182.736
0.016	0.063	0.082	87.071	87.685	86.457	182.412
0.016	0.063	0.086	86.283	86.893	85.672	182.074
0.031	0.031	0.008	92.255	92.775	91.735	188.935
0.031	0.031	0.012	91.704	92.218	91.190	188.667
0.031	0.031	0.016	91.359	91.885	90.833	188.384
0.031	0.031	0.020	90.983	91.495	90.471	188.116
0.031	0.031	0.024	90.709	91.247	90.172	187.856
0.031	0.031	0.027	90.315	90.821	89.808	187.557
0.031	0.031	0.031	89.925	90.460	89.391	187.234
0.031	0.031	0.035	89.545	90.081	89.009	186.981
0.031	0.031	0.039	89.185	89.774	88.595	186.656
0.031	0.031	0.043	88.453	89.009	87.896	186.367
0.031	0.031	0.047	88.101	88.698	87.505	186.065
0.031	0.031	0.051	87.490	88.070	86.910	185.791
0.031	0.031	0.055	87.043	87.594	86.491	185.447
0.031	0.063	0.039	92.098	92.588	91.608	191.192
0.031	0.063	0.043	91.565	92.093	91.037	190.980
0.031	0.063	0.047	91.186	91.715	90.658	190.721
0.031	0.063	0.051	90.844	91.375	90.313	190.486
0.031	0.063	0.055	90.505	91.085	89.925	190.262
0.031	0.063	0.059	90.131	90.658	89.604	190.054
0.031	0.063	0.063	89.816	90.323	89.309	189.791
0.031	0.063	0.067	89.459	90.020	88.898	189.560

Continued on next page

Table 9. ResNet50D results on TINYIMAGENET.

Delta1	Delta2	Delta3	accuracy_mean	accuracy_upper	accuracy_lower	setsize_mean
0.031	0.063	0.071	89.089	89.639	88.540	189.325
0.031	0.063	0.075	88.685	89.259	88.111	189.088
0.031	0.063	0.078	88.130	88.716	87.544	188.826
0.031	0.063	0.082	87.910	88.536	87.284	188.617
0.031	0.063	0.086	87.440	88.065	86.815	188.369
0.047	0.031	0.008	91.668	92.153	91.183	192.779
0.047	0.031	0.012	91.484	91.988	90.981	192.646
0.047	0.031	0.016	91.208	91.781	90.634	192.513
0.047	0.031	0.020	90.961	91.479	90.444	192.353
0.047	0.031	0.024	90.650	91.211	90.089	192.217
0.047	0.031	0.027	90.322	90.893	89.752	192.073
0.047	0.031	0.031	89.971	90.519	89.424	191.910
0.047	0.031	0.035	89.580	90.183	88.978	191.754
0.047	0.031	0.039	89.150	89.673	88.626	191.568
0.047	0.031	0.043	88.786	89.335	88.237	191.437
0.047	0.031	0.047	88.324	88.855	87.793	191.265
0.047	0.031	0.051	87.888	88.512	87.265	191.098
0.047	0.031	0.055	87.497	88.180	86.813	190.932
0.047	0.063	0.039	91.610	92.134	91.086	193.911
0.047	0.063	0.043	91.375	91.911	90.838	193.802
0.047	0.063	0.047	91.133	91.678	90.588	193.679
0.047	0.063	0.051	90.919	91.436	90.403	193.551
0.047	0.063	0.055	90.669	91.192	90.145	193.422
0.047	0.063	0.059	90.402	90.912	89.893	193.303
0.047	0.063	0.063	90.109	90.670	89.548	193.172
0.047	0.063	0.067	89.694	90.255	89.134	193.052
0.047	0.063	0.071	89.425	89.963	88.887	192.887
0.047	0.063	0.075	89.129	89.693	88.565	192.762
0.047	0.063	0.078	88.622	89.165	88.079	192.652
0.047	0.063	0.082	88.379	88.968	87.790	192.514
0.047	0.063	0.086	88.112	88.676	87.547	192.380
0.063	0.031	0.008	90.540	91.043	90.036	193.229
0.063	0.031	0.012	90.301	90.796	89.806	193.146
0.063	0.031	0.016	90.120	90.674	89.566	193.056
0.063	0.031	0.020	89.959	90.535	89.384	192.953
0.063	0.031	0.024	89.593	90.169	89.017	192.879
0.063	0.031	0.027	89.444	89.960	88.927	192.774
0.063	0.031	0.031	89.248	89.792	88.705	192.711
0.063	0.031	0.035	89.134	89.703	88.566	192.629
0.063	0.031	0.039	88.914	89.512	88.315	192.560
0.063	0.031	0.043	88.830	89.379	88.280	192.464
0.063	0.031	0.047	88.561	89.179	87.943	192.388
0.063	0.031	0.051	88.252	88.844	87.660	192.317
0.063	0.031	0.055	88.097	88.683	87.512	192.225
0.063	0.063	0.039	90.381	90.931	89.831	193.775
0.063	0.063	0.043	90.129	90.641	89.617	193.702
0.063	0.063	0.047	90.007	90.519	89.494	193.630
0.063	0.063	0.051	89.824	90.445	89.204	193.561
0.063	0.063	0.055	89.640	90.210	89.070	193.504
0.063	0.063	0.059	89.426	90.053	88.799	193.470
0.063	0.063	0.063	89.330	89.926	88.733	193.413
0.063	0.063	0.067	89.079	89.659	88.498	193.320
0.063	0.063	0.071	88.857	89.473	88.240	193.259
0.063	0.063	0.075	88.669	89.307	88.032	193.190
0.063	0.063	0.078	88.525	89.156	87.895	193.161
0.063	0.063	0.082	88.332	88.908	87.756	193.106
0.063	0.063	0.086	88.235	88.880	87.590	193.033

C.2. Figures for Resnet50d

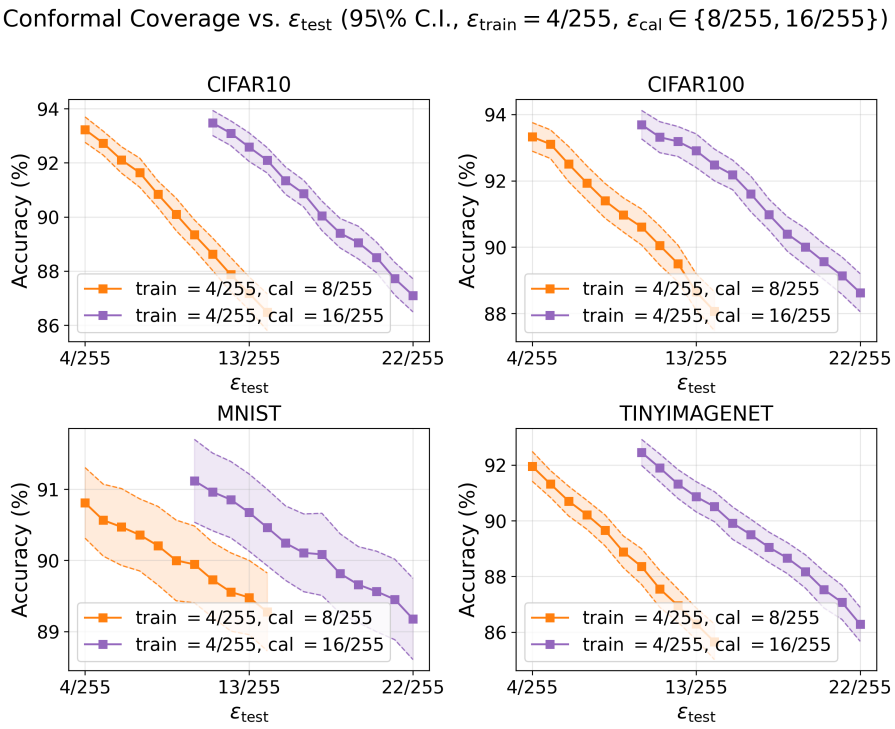

Figure 4. Resnet50d Accuracy

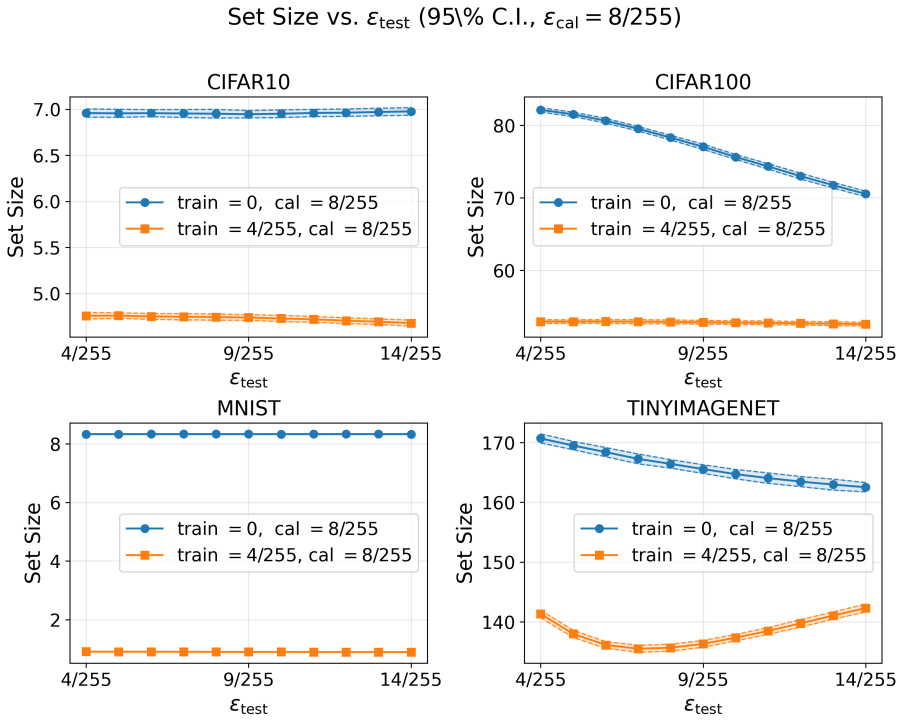

Figure 5. Resnet50d Set Size

ResNet50d. Across CIFAR10/100, MNIST, and TinyImageNet, the qualitative trends mirror our ViT results:

- *Accuracy vs. ϵ_{test}.* Accuracy decreases monotonically as ϵ_{test} increases, while a stronger calibration set ($\epsilon_{\text{cal}} = 16/255$ vs. $8/255$) shifts the curves upward and enlarges the ϵ_{test} range that meets the target marginal coverage band (95% CIs shown).
- *Effect of adversarial training on set size.* Under $\epsilon_{\text{train}} = 4/255$, adversarial training produces substantially smaller and more stable conformal prediction sets than clean training on CIFAR10/100 and MNIST, echoing the ViT trends.

Summary. ResNet50d reproduces the ViT conclusions: stronger calibration improves test-time robustness, and adversarial training stabilizes and shrinks conformal set sizes while maintaining comparable marginal coverage.

C.3. l_2 norm attack on ResNet50d

Train eps_train	Cal eps_cal	Mean acc. (%)	Worst acc. (%)	Mean set size	Max set size
0/255	136/255	90.14	89.44	6.47	7.20
128/255	136/255	90.30	89.32	4.00	4.50
132/255	136/255	90.46	89.38	3.88	4.68
136/255	136/255	90.30	88.68	3.85	4.33
140/255	136/255	90.40	89.30	3.83	4.16
144/255	136/255	90.31	89.78	3.90	4.46
0/255	140/255	90.21	89.24	6.29	6.76
128/255	140/255	90.06	88.88	3.79	4.24
132/255	140/255	90.38	89.58	3.85	4.26
136/255	140/255	90.32	89.66	3.88	4.46
140/255	140/255	90.25	89.40	3.85	4.63
144/255	140/255	90.23	89.52	3.90	4.23

Table 10. l_2 norm attack on ResNet50d.