How Learning Dynamics Drive Adversarially Robust Generalization?

Yuelin Xu[1] **Xiao Zhang**[1]

[1] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Abstract

Despite being widely adopted as a canonical framework for learning robust models, adversarial training suffers from robust overfitting. Existing empirical and theoretical explorations fail to provide a satisfactory mechanistic interpretation of the phenomenon. By modeling adversarial training with momentum SGD as a discrete-time dynamical system, we propose a PAC-Bayesian analytical framework that proves time-resolved robust generalization bounds. Specifically, our framework tracks the closed-form evolution of the posterior mean and covariance under both stationary and non-stationary transient regimes, connecting the model's robust generalization performance to learning rate, local loss geometry, and mini-batch stochastic gradients. By estimating the key quantities associated with the bound, we illustrate the underlying mechanism of robust overfitting. Our framework also shows how adversarial weight perturbation reduces robust generalization gaps by suppressing dominant loss-curvature modes, while suggesting that excessive penalization can be sub-optimal for optimization.

1 INTRODUCTION

Adversarial training [Madry et al., 2018], which uses projected gradient descent (PGD) to solve a minimax optimization objective, remains the most widely adopted algorithmic framework for training neural networks to be resilient to inputs crafted with small perturbations intended to fool the model [Szegedy et al., 2014]. Despite its prevalence, adversarial training exhibits a striking failure mode: robust test accuracy can deteriorate late in training—often immediately after a learning rate decay—even as the robust training loss continues to decrease. This phenomenon, known as *robust overfitting* [Rice et al., 2020], suggests that fitting the training objective does not necessarily translate into improved robust generalization on unseen data. To address the issue, various techniques have been proposed [Zhang et al., 2019, Wu et al., 2020, Carmon et al., 2019], yet a satisfying mechanistic interpretation of the phenomenon is lacking. While many heuristic robustness measures have been proposed, no single one can serve as a universal, reliable indicator of the model's robust generalization capability [Kim et al., 2023].

The surprising robust overfitting phenomenon and its complexity have motivated theoretical studies [Xiao et al., 2022, Wang et al., 2023, Xiao et al., 2023, Fu and Wang, 2025, Liu et al., 2024, Walter et al., 2026, Tian and Mao, 2025] to analyze the factors that influence adversarially robust generalization by adapting classical frameworks such as PAC-Bayes and algorithmic stability. Nevertheless, the worst-case robustness guarantees derived from these studies are often loose and rely on strong assumptions that may not reflect the actual behavior of algorithms used in practice. Moreover, these bounds are usually *static*, designed to certify a predictor at a fixed model checkpoint, but are insufficient to characterize the *time-varying dynamics* of adversarial training, which is essential to fully understand robust overfitting.

In this work, we argue that robust overfitting is driven by a transient imbalance between loss curvature and stochastic noise as the model navigates the adversarial loss landscape through stochastic gradient descent (SGD). To uncover the underlying mechanism, we adopt the PAC-Bayesian framework and model the SGD dynamics of adversarial training algorithms as a discrete-time dynamical system (Section 4.1). By treating the iterative parameter distribution as an implicit posterior, we derive closed-form solutions that characterize how its mean and covariance evolve with learning rate, local curvature, and gradient-noise dynamics, which yield schedule-aware, time-varying robust generalization bounds tailored for different training stages of interest (Section 4.2). Based on an efficient spectral estimation protocol, we empirically estimate the quantities identified by the theory by tracking the per-epoch Hessian and gradient-noise dynamics across different learning algorithms, thereby char-

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:7562–7593.

acterizing the driving factors associated with adversarially robust generalization (Section 5). Together, our analyses support a *time-varying mechanistic interpretation* of robust overfitting: a sharp learning-rate decay leads to posterior contraction along sharp directions, benefiting robust optimization while lowering the curvature-weighted variance during the initial transient phase (corresponding to the early jump of training and testing robust accuracy); however, the Hessian eigenvalues keep increasing as adversarial training prolongs, which eventually boosts the curvature-weighted variance despite the contracted posterior and adversely affects the model's robust generalization (corresponding to the gradual degradation of robust test accuracy in later epochs).

Our main contributions are summarized as follows:

- By modeling momentum SGD as a dynamical system, we prove PAC-Bayesian robust generalization bounds for different learning stages of adversarial training. As a byproduct, our method can explicitly track the time-varying evolution of the posterior mean and covariance.
- Through empirical estimation of the spectral quantities reflected in the bound, we provide a time-resolved diagnostic account of robust overfitting, illustrating how learning-rate decay, progressive sharpening, and posterior contraction jointly shape robust generalization.
- We conduct controlled experiments to analyze the diagnostic patterns across various learning algorithms, suggesting that adversarial weight perturbation [Wu et al., 2020] may overly penalize Hessian eigenvalues.

2 RELATED WORK

Robust Overfitting. Since the initial discovery of adversarial examples [Szegedy et al., 2014], numerous methods have been proposed to improve model robustness [Goodfellow et al., 2015, Papernot et al., 2016, Wong and Kolter, 2018, Cohen et al., 2019]. Among them, *adversarial training* (AT), a minimax optimization framework that employs *projected gradient descent* (PGD), is most popular [Madry et al., 2018]. However, AT exhibits a recurring failure mode: robust test accuracy can drop late in training—often right after learning rate decay—even as the robust training loss continues to decrease, a phenomenon termed *robust overfitting* [Rice et al., 2020]. Empirical remedies span three main categories: objective-based methods like TRADES [Zhang et al., 2019, Chen et al., 2020] redesign the training optimization objective, geometry-aware approaches like *adversarial weight perturbation* (AWP) [Liu et al., 2020, Wu et al., 2020, Foret et al., 2021] penalize the model's sharpness for finding flatter solutions, and augmentation-based schemes [Carmon et al., 2019, Gowal et al., 2021] incorporate extra data in adversarial training. Despite improvements, the above techniques are all empirical, which lacks a unified, mechanistic interpretation of the robust overfitting phenomenon.

Adversarially Robust Generalization. Existing literature has analyzed robust generalization using various theoretical tools. A line of work adapted the classical PAC-Bayes framework [McAllester, 1999, Neyshabur et al., 2018, Bartlett et al., 2021] to derive posterior-based robust generalization bounds [Mustafa et al., 2023, 2024, Xiao et al., 2023, Wang et al., 2023], often relating the KL-divergence complexity term to flatness-based measures. Another line of research studied adversarially robust generalization from the perspective of algorithmic stability [Xing et al., 2021, Xiao et al., 2022, Tian and Mao, 2025], which captures a learning algorithm's insensitivity to small perturbations in the training dataset. Additionally, Fu and Wang [2025] extended neural tangent kernel (NTK) analysis to AT, deriving closed-form dynamics under squared loss for neural networks with infinite width, while Liu et al. [2024] focused on characterizing the impact of individual training instances on robust overfitting. Despite rigorous proofs, most theories rely on assumptions that deviate from practice. Even when assumptions are acceptable, the resulting bounds are often loose and cannot tightly explain the empirical robust overfitting curves. These observations highlight a critical limitation: existing studies focus on *static* worst-case guarantees; however, robust generalization requires understanding the model's internal dynamics, suggesting the need for better analytical tools to characterize the actual adversarial training dynamics.

SGD Dynamics. Recently, a line of literature has proposed studying standard deep learning generalization by modeling SGD as a discrete-time stochastic dynamical system [Liu et al., 2021, Ziyin et al., 2022, 2024]. In particular, they showed how the learning rate, loss curvature, and gradient noise structure jointly shape parameter fluctuations, which can be used to derive an exact analytical form of the stationary distribution and explain how SGD escapes from sharp minima. Compared with prior literature that uses continuous-time SDEs or Langevin approximations [Mandt et al., 2017, Li et al., 2017], this line of work accounts for the learning step size and minibatch stochastic noise, providing more realistic predictions for SGD dynamics and implicit regularization. We leverage recent advances in modeling SGD dynamics to characterize *time-resolved* PAC-Bayesian robust generalization bounds for adversarial training, offering a principled framework for understanding robust overfitting.

3 PRELIMINARIES

Adversarial Risk. We work with the following theoretical definition of adversarial risk, which often serves as the basis for evaluating an ML model's robustness against worst-case input perturbations [Madry et al., 2018, Rice et al., 2020].

Definition 3.1 (Adversarial risk)**.** Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space, $\mathcal{Y}$ be the set of class labels, and $\mathcal{D}$ be a probability distribution over $\mathcal{X} \times \mathcal{Y}$. For any model $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ with parameters $\boldsymbol{w} \in \mathbb{R}^m$, the *adversarial risk* of $f_{\boldsymbol{w}}$ over $\mathcal{D}$

against L_p perturbations bounded by $\epsilon \geq 0$ is defined as:

$$\mathcal{R}_\epsilon(\boldsymbol{w}) = \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\exists\, \boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x}), f_{\boldsymbol{w}}(\boldsymbol{x}') \neq y\big], \quad (1)$$

where $\mathcal{B}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}+\boldsymbol{\delta} \mid \boldsymbol{\delta} \in \mathbb{R}^d, \|\boldsymbol{\delta}\|_p \leq \epsilon\}$ stands for the perturbation ball centered at $\boldsymbol{x}$ with radius ϵ in L_p-norm.

Note that when $\epsilon = 0$, adversarial risk reduces to the standard notion of risk. In line with the existing literature, we focus on the most widely considered L_p perturbations. Correspondingly, the adversarial robustness of an ML model $f_{\boldsymbol{w}}$ can be understood as $1 - \mathcal{R}_\epsilon(\boldsymbol{w})$. According to Equation 1, one can show that adversarial risk is mathematically equivalent to $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\max_{\|\boldsymbol{\delta}\|_p\leq\epsilon} \mathbb{1}\{f_{\boldsymbol{w}}(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\}]$.

Adversarial Training. To effectively train robust models with low adversarial risk, adversarial training is the most popular framework [Goodfellow et al., 2015, Madry et al., 2018]. Given a set of examples $\mathcal{S}$ i.i.d. sampled from the underlying distribution $\mathcal{D}$, adversarial training is designed to minimize the following *empirical adversarial loss*:

$$\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}) = \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}}\left[\max_{\|\boldsymbol{\delta}\|_p\leq\epsilon} \ell(\boldsymbol{w}, \boldsymbol{x}+\boldsymbol{\delta}, y)\right], \quad (2)$$

where $\ell(\boldsymbol{w}, \boldsymbol{x}+\boldsymbol{\delta}, y)$ stands for surrogate loss, such as hinge or cross-entropy loss, measured at perturbed inputs. Compared with the non-differentiable objective of adversarial risk (Equation 1), using a surrogate loss is more amenable to optimization. Correspondingly, we define the *expected adversarial loss* over the entire data distribution $\mathcal{D}$ as:

$$\mathcal{L}_\epsilon(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\|\boldsymbol{\delta}\|_p\leq\epsilon} \ell(\boldsymbol{w}, \boldsymbol{x}+\boldsymbol{\delta}, y)\right]. \quad (3)$$

In the existing literature, PGD is commonly used to optimize the inner maximization problem [Madry et al., 2018], while SGD with momentum [Sutskever et al., 2013] is often adopted to optimize the model parameters. Formally, the iterative model updates can be cast as: for $t = 0, 1, \ldots,$

$$\boldsymbol{v}_{t+1} = \kappa\boldsymbol{v}_t + \nabla\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_t) + \boldsymbol{\xi}_t,\ \boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\boldsymbol{v}_{t+1}, \quad (4)$$

where $\eta > 0$ denotes the SGD learning rate, $\kappa \in [0, 1)$ is the momentum hyperparameter, $\boldsymbol{v}_t$ denotes the velocity vector (accumulated momentum), and $\boldsymbol{\xi}_t$ stands for the difference at iterate t between the stochastic gradient with respect to a random mini-batch and the full gradient $\nabla\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_t)$. At the initial state, $\boldsymbol{w}_0$ corresponds to the initial model parameters, usually according to some random distribution, and $\boldsymbol{v}_0 = \boldsymbol{0}$.

PAC-Bayesian Robust Generalization Bound. The PAC-Bayesian framework has been pivotal in analyzing the generalization of ML models [McAllester, 1999, Neyshabur et al., 2018, Dziugaite and Roy, 2017, Xiao et al., 2023, Alquier et al., 2024]. The following theorem, proven in Appendix C.1, illustrates how to bound the expected adversarial loss over any posterior using the PAC-Bayesian framework.

Theorem 3.2 (PAC-Bayesian robust generalization bound for bounded losses)**.** *Let $\mathcal{S}$ be a set of examples i.i.d. sampled from $\mathcal{D}$ and $\mathcal{W} \subseteq \mathbb{R}^m$ be the space of model parameters. Suppose $\mathcal{P}$ is a data-independent prior over $\mathcal{W}$, and $\mathcal{Q}$ is a posterior distribution supported on $\mathcal{W}$. For any $\beta > 0$ and $\alpha \in (0, 1)$, with probability at least $1-\alpha$, we have*

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\mathcal{L}_\epsilon(\boldsymbol{w})\big] \leq \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] + \frac{1}{\beta}\mathrm{KL}(\mathcal{Q}\,\|\,\mathcal{P}) + \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta}\ln\alpha, \quad (5)$$

where C_ℓ is a scalar upper bound on the loss function[1].

Theorem 3.2 suggests that to achieve robust generalization, $\mathcal{Q}$ needs to be selected such that it can fit the empirical data well (low $\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})]$) and the KL divergence between $\mathcal{Q}$ and the prior $\mathcal{P}$ is small. To obtain a tighter bound, β is usually set to $O\big(\sqrt{|\mathcal{S}|}\big)$ for balancing the terms. Note that Equation 5 holds for any data-independent prior and posterior distributions; the generality of the PAC-Bayesian bound enables us to later analyze the model's robust generalization performance at different stages during adversarial training.

4 HOW LEARNING DYNAMICS SHAPE ROBUST GENERALIZATION?

4.1 LINK TO LOSS & POSTERIOR GEOMETRY

To further digest Theorem 3.2, we assume that the prior $\mathcal{P}$ and the posterior $\mathcal{Q}$ follow Gaussian distributions (Lemma 4.1), and the adversarial loss $\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$ can be locally upper bounded via second-order Taylor expansion (Lemma 4.2). Both assumptions have been widely adopted in the prior PAC-Bayesian literature for closed-form analysis [Germain et al., 2009, Bartlett et al., 2021, Liu et al., 2021, Alquier et al., 2024]. Based on these assumptions, Equation 5 can then be reduced to a more manageable form (Theorem 4.3).

Lemma 4.1. *Assume the prior $\mathcal{P}$ and the posterior $\mathcal{Q}$ defined in Theorem 3.2 follow Gaussian distributions:*

$$\mathcal{P} = \mathcal{N}(\boldsymbol{0}, \sigma_{\mathcal{P}}^2\mathbf{I}),\ \textit{and}\ \ \mathcal{Q} = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\Sigma}_{\mathcal{Q}}), \quad (6)$$

where $\sigma_{\mathcal{P}} \in \mathbb{R}_+$, $\boldsymbol{\mu}_{\mathcal{Q}} \in \mathbb{R}^m$, $\boldsymbol{\Sigma}_{\mathcal{Q}} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Sigma}_{\mathcal{Q}} \succ 0$. Then, we can derive the following closed-form expression:

$$\mathrm{KL}(\mathcal{Q}\,\|\,\mathcal{P}) = \frac{1}{2\sigma_{\mathcal{P}}^2}\big(\|\boldsymbol{\mu}_{\mathcal{Q}}\|_2^2 + \mathrm{Tr}(\boldsymbol{\Sigma}_{\mathcal{Q}})\big) - \frac{1}{2}\ln\det(\boldsymbol{\Sigma}_{\mathcal{Q}}) + \frac{m}{2}\ln\sigma_{\mathcal{P}}^2 - \frac{m}{2}. \quad (7)$$

We present the proof of Lemma 4.1 in Appendix A.1. Compared to the spherical Gaussian posteriors adopted in literature [Germain et al., 2009, Jin et al., 2022, Alquier et al.,

[1] For cross-entropy loss, one can derive similar concentration results by applying PAC-Bayes-Chernoff bounds [Casado et al., 2024] for unbounded losses under light-tail or moment conditions.

2024], our choice of a general covariance $\mathbf{\Sigma}_{\mathcal{Q}}$ enables an analytically tractable yet less restrictive bound that captures anisotropic parameter variability. Note that Lemma 4.1 and its proof can be extended to a more general scenario where $\mathcal{Q}$ is modeled as a mixture of Gaussians (Corollary A.1).

To deal with the expected adversarial loss in Equation 5, we introduce a local quadratic loss assumption, which connects the term to the gradient, Hessian, and posterior structure.

Lemma 4.2. *For a given posterior $\mathcal{Q}$, assume there exists a reference point $\tilde{\boldsymbol{w}} \in \mathbb{R}^m$ such that, on the local region carrying the mass of $\mathcal{Q}$, the empirical loss $\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$ satisfies:*

$$\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}) \leq \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \langle \nabla \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}), \Delta \boldsymbol{w} \rangle + \frac{1}{2} \Delta \boldsymbol{w}^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) \Delta \boldsymbol{w} + r_{\tilde{\boldsymbol{w}}}(\boldsymbol{w}), \tag{8}$$

where $\Delta \boldsymbol{w} = \boldsymbol{w} - \tilde{\boldsymbol{w}}$, $\nabla \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}})$ is the gradient of the empirical adversarial loss, $\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})$ is the Hessian matrix, and $r_{\tilde{\boldsymbol{w}}}(\boldsymbol{w})$ is the higher-order remainder term. Then we have

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] \leq \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \langle \nabla \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}), \Delta \bar{\boldsymbol{w}} \rangle + \frac{1}{2} \Delta \bar{\boldsymbol{w}}^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) \Delta \bar{\boldsymbol{w}} + \frac{1}{2} \mathrm{Tr}\big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) \mathbf{\Sigma}_{\mathcal{Q}}\big) + R_{\mathcal{Q}}(\tilde{\boldsymbol{w}}), \tag{9}$$

where $\Delta \bar{\boldsymbol{w}} = \boldsymbol{\mu}_{\mathcal{Q}} - \tilde{\boldsymbol{w}}$ is the posterior mean drift from the reference point $\tilde{\boldsymbol{w}}$ and $R_{\mathcal{Q}}(\tilde{\boldsymbol{w}}) = \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[r_{\tilde{\boldsymbol{w}}}(\boldsymbol{w})]$.

Here, the reference point $\tilde{\boldsymbol{w}}$ can be understood as the local coordinate anchor for the posterior being analyzed. Once the posterior is fixed, $\tilde{\boldsymbol{w}}$ is chosen in the same local basin or short-time window such that $\boldsymbol{w} - \tilde{\boldsymbol{w}}$ is small for most posterior mass. Lemma 4.2, with its detailed proof provided in Appendix A.2, illustrates that the posterior-averaged empirical adversarial loss is regulated by the local geometry of the loss landscape and the posterior covariance. Similar decomposition extends to Gaussian-mixture posteriors by applying the expansion within each component (Corollary A.1).

Compared with the standard loss, the inner maximization operator unique to the adversarial loss (Equation 2) complicates the derivation of the Hessian $\hat{\mathbf{H}}_\epsilon$. Adopting a local quadratic loss decomposition will largely simplify our later analysis while capturing the leading variations. Strictly speaking, $\hat{\mathbf{H}}_\epsilon$ may not exist everywhere; however, note that standard neural networks are usually piecewise smooth (e.g., ReLU-based), the adversarial loss can thus be considered a maximum over piecewise-smooth functions and is therefore twice differentiable almost everywhere. This implies that Equation 9 holds for almost all inputs and parameters. In our experiments, we adopt the common approach to estimate the Hessian by ignoring the second-order dependence of $\boldsymbol{\delta}^*$ on $\boldsymbol{w}$ using a Danskin-type argument and employing PGD to approximate the inner maximization [Liu et al., 2020].

Equipped with Lemmas 4.1 and 4.2, we can now reduce the generic PAC-Bayesian robust generalization bound in Theorem 3.2 to the following theorem proven in Appendix A.3.

Theorem 4.3. *Let $\mathcal{P} = \mathcal{N}(\mathbf{0}, \sigma_{\mathcal{P}}^2 \mathbf{I})$ and $\mathcal{Q}$ be a posterior that satisfies the conditions in Lemmas 4.1 and 4.2. For any $\beta > 0$, $\alpha \in (0, 1)$, with probability at least $1 - \alpha$, we have*

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\mathcal{L}_\epsilon(\boldsymbol{w})\big] \leq \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \big\langle \nabla \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}), \Delta_{\mathcal{Q}} \big\rangle + \frac{1}{2} \Delta_{\mathcal{Q}}^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) \Delta_{\mathcal{Q}} + \frac{1}{2} \mathrm{Tr}\big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) \mathbf{\Sigma}_{\mathcal{Q}}\big) + R_{\mathcal{Q}}(\tilde{\boldsymbol{w}}) + \frac{1}{2\beta\sigma_{\mathcal{P}}^2} \big(\|\boldsymbol{\mu}_{\mathcal{Q}}\|_2^2 + \mathrm{Tr}(\mathbf{\Sigma}_{\mathcal{Q}})\big) - \frac{1}{2\beta} \ln\det(\mathbf{\Sigma}_{\mathcal{Q}}) + C_{\mathrm{PB}}, \tag{10}$$

where $\Delta_{\mathcal{Q}} = \boldsymbol{\mu}_{\mathcal{Q}} - \tilde{\boldsymbol{w}}$, and C_{PB} collects all the terms that are independent of the posterior $\mathcal{Q}$, formally defined as:

$$C_{\mathrm{PB}} = \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta} \ln \alpha + \frac{m}{2\beta} \ln \sigma_{\mathcal{P}}^2 - \frac{m}{2\beta},$$

and C_ℓ is a constant introduced in Theorem 3.2.

Theorem 4.3 connects the PAC-Bayesian robust generalization bound to the internal adversarial training dynamics (see Corollary A.1 for the extension to Gaussian mixtures). Specifically, the expected adversarial loss over the posterior is controlled by three main components: (i) first- and second-order biases, (ii) curvature-weighted variance, and (iii) KL-divergence related terms, capturing the interactions between the local geometry of the loss landscape (gradient and Hessian) and the posterior (mean drift and covariance). During adversarial training, the local geometry of the loss landscape and posterior can vary drastically, necessitating a more fine-grained temporal analysis of their evolution.

4.2 TIME-VARYING POSTERIOR DYNAMICS

While Theorem 4.3 relates robust generalization bounds to specific local geometry-induced properties, the posterior-related terms remain obscure. In this section, we illustrate how to use dynamical system modeling techniques to derive closed-form solutions of the posterior dynamics under both stationary and non-stationary transient regimes.

Stationary Regime. Let $\mathcal{Q}_t$ denote the posterior distribution at timestep t induced by the momentum SGD update rule (Equation 4). We say the learning system reaches a *stationary* state at time t_0 if $\mathcal{Q}_t = \mathcal{Q}_{t_0}$ for any $t \geq t_0$. The following lemma, proven in Appendix B.1, characterizes the state-space representation of the posterior dynamics, as well as the associated mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ at stationarity.

Lemma 4.4. *Suppose an ML system defined by Equation 4 reaches a stationary state at time t_0. Denote by the joint state vector $\boldsymbol{u}_t = [\boldsymbol{w}_t - \tilde{\boldsymbol{w}}; \boldsymbol{v}_t] \in \mathbb{R}^{2m}$. Under the local quadratic loss assumption as in Lemma 4.2, we have*

$$\forall t \geq t_0, \;\; \boldsymbol{u}_{t+1} = \mathbf{A} \boldsymbol{u}_t + \mathbf{G}\big[\nabla \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \boldsymbol{\xi}_t\big], \tag{11}$$

where $\boldsymbol{\xi}_t$ is the mini-batch gradient noise at time t. Assume the mini-batch noise forms a martingale-difference sequence

conditioned on the training set, $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathcal{F}_t, \mathcal{S}] = \mathbf{0}$, where $\mathcal{F}_t$ is the filtration generated by the past training trajectory, and $\mathbf{A}$ and $\mathbf{G}$ are the transition matrix and bias vector of the dynamical system, respectively defined as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} - \eta\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) & -\eta\kappa\mathbf{I} \\ \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) & \kappa\mathbf{I} \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} -\eta\mathbf{I} \\ \mathbf{I} \end{bmatrix}. \quad (12)$$

Let $\mathbf{C} = \lim_{t\to\infty} \mathbb{E}[\boldsymbol{\xi}_t\boldsymbol{\xi}_t^\top \mid \mathcal{S}]$ denote the finite stationary gradient-noise covariance. Assume $\mathbf{H}_{\mathrm{ref}} := \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}) \succ 0$, $\mathbf{C}$ commutes with $\mathbf{H}_{\mathrm{ref}}$, and every eigenvalue λ_i of $\mathbf{H}_{\mathrm{ref}}$ satisfies $0 < \eta\lambda_i < 2(1+\kappa)$, $i = 1, 2, \ldots, m$. Then, we can derive the stationary posterior mean and covariance:

$$\boldsymbol{\mu} = \tilde{\boldsymbol{w}} - \left[\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\right]^{-1}\nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}), \quad (13)$$

$$\boldsymbol{\Sigma} = \left[\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\left(2\mathbf{I} - \frac{\eta}{1+\kappa}\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\right)\right]^{-1}\frac{\eta}{1-\kappa}\mathbf{C}. \quad (14)$$

Lemma 4.4 reveals two key properties regarding the stationary posterior. First, Equation 13 indicates that the stationary mean involves a curvature-weighted gradient correction with respect to the reference $\tilde{\boldsymbol{w}}$. When translated into first- and second-order bias terms, it yields a Newton-type optimization gain ($-\frac{1}{2}\nabla\hat{\mathcal{L}}_\epsilon^\top\hat{\mathbf{H}}_\epsilon^{-1}\nabla\hat{\mathcal{L}}_\epsilon$), acting as a constant force for lowering the empirical adversarial loss within the stationary trajectory. Second, Equation 14 makes explicit how learning rate η, Hessian $\hat{\mathbf{H}}_\epsilon$, and noise covariance $\mathbf{C}$ jointly shape the posterior spread: for each eigendirection of $\hat{\mathbf{H}}_\epsilon$, larger eigenvalues accelerate the contraction while larger noise variances and learning rates broaden it. To ensure the covariance is well defined at stationarity, the Hessian eigenvalues need to stay within the range of $0 < \lambda_i < 2(1+\kappa)/\eta$. This stability condition implicitly imposes a regularization effect on the loss curvature when the learning rate η is large (prior to the first learning-rate decay in adversarial training).

Non-Stationary Regime. Although stationary analysis is the primary focus for analyzing SGD dynamics in prior literature [Liu et al., 2021, Ziyin et al., 2022], it is insufficient to capture the non-stationary transition during the onset of robust overfitting, where the learning rate sharply decays and the system drifts away from its previous stationary state. Note that the transition matrix and noise covariance depend on the local loss geometry with reference to $\tilde{\boldsymbol{w}}$, which can vary across time under non-stationary regimes. To obtain a tractable approximation, we assume the system remains approximately invariant over short time windows and characterize the posterior dynamics via iterative linearization.

Lemma 4.5. *Let $\mathcal{Q}_t$ be the posterior induced by the SGD update rule in Equation 4 at timestep t. Assume that the learning rate η_t and Hessian $\mathbf{H}_t := \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)$ remain fixed over the next k updates. Assume that $\{\boldsymbol{\xi}_s\}_{s=t}^{t+k-1}$ is a martingale-difference sequence conditional on $\mathcal{S}$, $\mathbb{E}[\boldsymbol{\xi}_s \mid \mathcal{F}_s, \mathcal{S}] = \mathbf{0}$, and that its conditional covariance is fixed at $\mathbf{C}_t$ within the window, where $\mathbf{C}_t$ is finite and $0 < \eta_t\lambda_{t,i} < 2(1+\kappa)$ for every analyzed mode. Consider the joint state vector $\boldsymbol{u}_t = [\boldsymbol{w}_t - \tilde{\boldsymbol{w}}_t; \boldsymbol{v}_t]$ and let $\boldsymbol{\Omega}_t = \mathrm{Cov}(\boldsymbol{u}_t)$ be its covariance. Then, at timestep $t' = t + k$, the posterior $\mathcal{Q}_{t'}$ satisfies:*

$$\mathbb{E}[\boldsymbol{u}_{t'}] = \mathbf{A}_t^k\,\mathbb{E}[\boldsymbol{u}_t] + \sum\nolimits_{j=0}^{k-1}\mathbf{A}_t^j\mathbf{G}\nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t), \quad (15)$$

$$\boldsymbol{\Omega}_{t'} = \mathbf{A}_t^k\boldsymbol{\Omega}_t(\mathbf{A}_t^k)^\top + \sum\nolimits_{j=0}^{k-1}\mathbf{A}_t^j\mathbf{G}\mathbf{C}_t\mathbf{G}^\top(\mathbf{A}_t^j)^\top. \quad (16)$$

Here, $\mathbf{A}_t$ denotes transition matrix, $\mathbf{C}_t = \mathbb{E}[\boldsymbol{\xi}_s\boldsymbol{\xi}_s^\top \mid \mathcal{F}_s, \mathcal{S}]$ is the conditional noise covariance, and $\mathbf{G}$ is the bias vector.

Lemma 4.5, proven in Appendix B.2, illustrates how the posterior evolves within the time window $[t, t']$, where the mean drift and covariance at time t' can be derived by projecting onto the corresponding state space: $\boldsymbol{\mu}_{t'} = \boldsymbol{\Pi}\mathbb{E}[\boldsymbol{u}_{t'}]$ and $\boldsymbol{\Sigma}_{t'} = \boldsymbol{\Pi}\boldsymbol{\Omega}_{t'}\boldsymbol{\Pi}^\top$, where $\boldsymbol{\Pi} = [\mathbf{I}\ \ \mathbf{0}]$. It is reasonable to assume the transition matrix is time-invariant over a short time period, as the learning rate η is often very small when robust overfitting happens for adversarial training algorithms, suggesting that the system is slowly evolving during the transient regime. To approximate the posterior dynamics at an arbitrary timestep, one can apply Lemma 4.5 to a sequence of consecutive time windows to accumulate the mean drift and posterior covariance fluctuation.

Final Robust Generalization Bounds. Now that we have derived the closed-form solutions for the stationary posterior and its evolution during the non-stationary transient stage, we can formalize the PAC-Bayesian robust generalization bound into the following theorem, proven in Appendix B.3.

Theorem 4.6. *Consider the dynamical system defined by the momentum SGD update rule in Equation 4. Let $\mathcal{Q}_t$ be the posterior at timestep t with mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$. Let $\boldsymbol{g}_t = \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t)$, $\mathbf{H}_t = \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)$, and $\Delta_t = \boldsymbol{\mu}_t - \tilde{\boldsymbol{w}}_t$. Suppose the assumptions in Lemmas 4.1 and 4.2 are satisfied; in particular, $\sigma_{\mathcal{P}} > 0$ and $\boldsymbol{\Sigma}_t \succ 0$. For any $\beta > 0$ and $\alpha \in (0, 1)$, with probability at least $1 - \alpha$, we have*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}_t}\left[\mathcal{L}_\epsilon(\boldsymbol{w})\right] \le\ & \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t) + \left\langle \boldsymbol{g}_t, \Delta_t\right\rangle \\ & + \frac{1}{2}\Delta_t^\top\mathbf{H}_t\Delta_t + \frac{1}{2}\,\mathrm{Tr}(\mathbf{H}_t\boldsymbol{\Sigma}_t) + \frac{1}{2\beta\sigma_{\mathcal{P}}^2}\,\mathrm{Tr}(\boldsymbol{\Sigma}_t) \\ & + \frac{1}{2\beta\sigma_{\mathcal{P}}^2}\|\boldsymbol{\mu}_t\|_2^2 - \frac{1}{2\beta}\ln\det(\boldsymbol{\Sigma}_t) + R_t + C_{\mathrm{PB}}, \quad (17)\end{aligned}$$

where $R_t = R_{\mathcal{Q}_t}(\tilde{\boldsymbol{w}}_t)$ and C_{PB} are defined in Lemma 4.2 and Theorem 4.3, respectively.

Plugging the derivations of posterior mean and covariance in Lemma 4.4 (stationary) or Lemma 4.5 (non-stationary transient) into Theorem 4.6, we can analyze each term associated with the PAC-Bayesian robust generalization bounds and how they evolve during adversarial training. It is worth noting that closed-form bound component derivations make use of the assumption that the Hessian $\mathbf{H}_t$ and the covariance matrices ($\mathbf{C}_t$ and $\boldsymbol{\Sigma}_t$) satisfy the commutativity condition (see Figure 3 in Appendix F.1 for supporting

Algorithm 1 Spectral estimation of bound components

Input: model checkpoint $\boldsymbol{w}$, rank k, hyperparameters used for PGD attack, number of sampled mini-batches M.
Generate adversarial mini-batches and compute gradients and Hessian–vector products of $\hat{\mathcal{L}}_\epsilon$ at $\boldsymbol{w}$.
Estimate top-k Hessian eigenpairs $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^k$ by power iteration using Hessian–vector products.
Let $\mathbf{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$. For each mini-batch $b \in \{1, \ldots, M\}$, compute $\boldsymbol{g}_b$ and $\boldsymbol{p}_b = \mathbf{V}^\top \boldsymbol{g}_b$.
Compute $\Gamma = \mathrm{Cov}_b(\boldsymbol{p}_b)$ and set $\gamma_i = \Gamma_{ii}$.
Compute $(\sigma_i^2, \Delta\bar{\boldsymbol{w}}_i)$ using stationary formulas (Lemma 4.4) or non-stationary transient recursions (Lemma 4.5).
Form the variance/entropy proxies and bias terms by summing contributions over $i \le k$.
Output: $\{\lambda_i\}$, $\{\gamma_i\}$, $\{\sigma_i^2\}$, and decomposed bound components.

experiments), suggesting that they can be simultaneously diagonalized. Specifically, we can show that the trace and determinant terms in Equation 17 can be computed as:

$$\frac{1}{2}\mathrm{Tr}(\mathbf{H}_t\boldsymbol{\Sigma}_t) = \frac{1}{2}\sum\nolimits_{i=1}^{m} \lambda_{t,i}\,\sigma_{t,i}^2,$$
$$\frac{1}{2\beta\sigma_{\mathcal{P}}^2}\mathrm{Tr}(\boldsymbol{\Sigma}_t) = \frac{1}{2\beta\sigma_{\mathcal{P}}^2}\sum\nolimits_{i=1}^{m} \sigma_{t,i}^2,$$
$$-\frac{1}{2\beta}\ln\det(\boldsymbol{\Sigma}_t) = -\frac{1}{2\beta}\sum\nolimits_{i=1}^{m} \ln\sigma_{t,i}^2,$$

where $\lambda_{t,i}$ and $\sigma_{t,i}^2$ denote the i-th eigenvalues of $\mathbf{H}_t$ and $\boldsymbol{\Sigma}_t$, respectively. In our experiments, we first estimate the top-k Hessian eigenvalues for each intermediate model and the corresponding projected noise-covariance eigenvalues, then approximate each relevant term[2] in the derived bounds.

5 EXPERIMENTS

We focus our main experiments on CIFAR-10 and L_∞ perturbations with $\epsilon = 8/255$ to study the robust generalization bounds, where we compare three learning algorithms: standard training (ST), adversarial training (AT) [Madry et al., 2018], and adversarial weight perturbation (AWP) [Wu et al., 2020], all trained on a PreActResNet-18 architecture for 200 epochs using momentum SGD ($\kappa = 0.9$) with batch size 128 and weight decay 5×10^{-4}. Aligned with Rice et al. [2020], we adopt a piecewise scheduler with an initial learning rate of 0.1, decayed to 0.01 at epoch 100 and to 0.001 at epoch 150. For AT and AWP, we evaluate the robust losses and errors with the intermediate models at each epoch, while we record the clean counterparts for ST (see Appendix D for details). Appendix F.3 reports all the additional experiments, where we vary datasets (CIFAR-100, SVHN, Imagenette-160), algorithms (TRADES, semi-supervised AT), perturbation radii, batch sizes, and learning-rate schedules.

5.1 SPECTRAL ESTIMATION

A computational challenge is efficiently estimating the statistical quantities, including the loss-geometry-related gradient and Hessian terms, as well as the posterior mean and covariance. Since the number of parameters m is huge for modern deep neural networks, computing the full Hessian matrix at a single checkpoint is very expensive. Therefore, we adopt a more efficient *per-epoch spectral estimation* protocol based on power iterations and Hessian-vector products to approximate the top-k eigensubspace with respect to the model checkpoint, to capture the Hessian's important structure while reducing computational cost. Accordingly, other quantities are projected onto the same subspace and then incorporated to obtain an estimate for each decomposed term of the robust generalization bounds in Theorem 4.6. Algorithm 1 demonstrates the adopted estimation procedure.

In particular, at each epoch t, we first estimate the top-k Hessian eigenpairs $\{(\lambda_i(t), v_i(t))\}_{i=1}^k$ with respect to the empirical loss by power iteration using Hessian-vector products. Note that for adversarial training algorithms, eigenpairs are computed with respect to the empirical PGD-based adversarial loss, whereas the unperturbed clean loss is used for standard training. Next, we approximate the associated gradient-noise covariances by sampling mini-batch gradients $\boldsymbol{g}_b(t)$ and projecting them onto the same eigensubspace $\mathbf{V}_t = [\boldsymbol{v}_1(t), \ldots, \boldsymbol{v}_k(t)]$. Eventually, we obtain $\{\gamma_i(t)\}_{i=1}^k$, where $\gamma_i(t)$ is the diagonal entry of the projected mini-batch gradient-noise covariance in the i-th Hessian eigendirection.

These estimates $\{\lambda_i(t), \gamma_i(t)\}_{i=1}^k$ will be used to compute the associated posterior variances $\{\sigma_i^2\}_{i=1}^k$, which capture the parameter fluctuations along each top-k eigendirection, through the stationary covariance expression (Equation 14) and the post-decay transient recursion (Equation 16). We adopt a similar scheme to approximate the posterior mean drift at stationarity (Equation 13) and the non-stationary variants (Equation 15). Finally, all the above estimates will be translated into Equation 17 to characterize the role of each decomposition in the derived PAC-Bayesian bounds.

Detailed estimation steps, including batch sampling and power-iteration configurations, and an analysis of its stability with respect to k are provided in Appendix D.2. Our estimation protocol is an offline checkpoint diagnostic, not a training-time method: it avoids forming the full Hessian and full noise covariance, and its cost scales with the number of evaluated checkpoints, retained modes, power iterations, and sampled mini-batches (see Appendix D.4 for details).

[2]We ignore $\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t)$, R_t and C_{PB} in our later diagnostic analysis, since they are either independent of $\mathcal{Q}$ or have negligible impact on robust generalization gap compared to remaining terms.

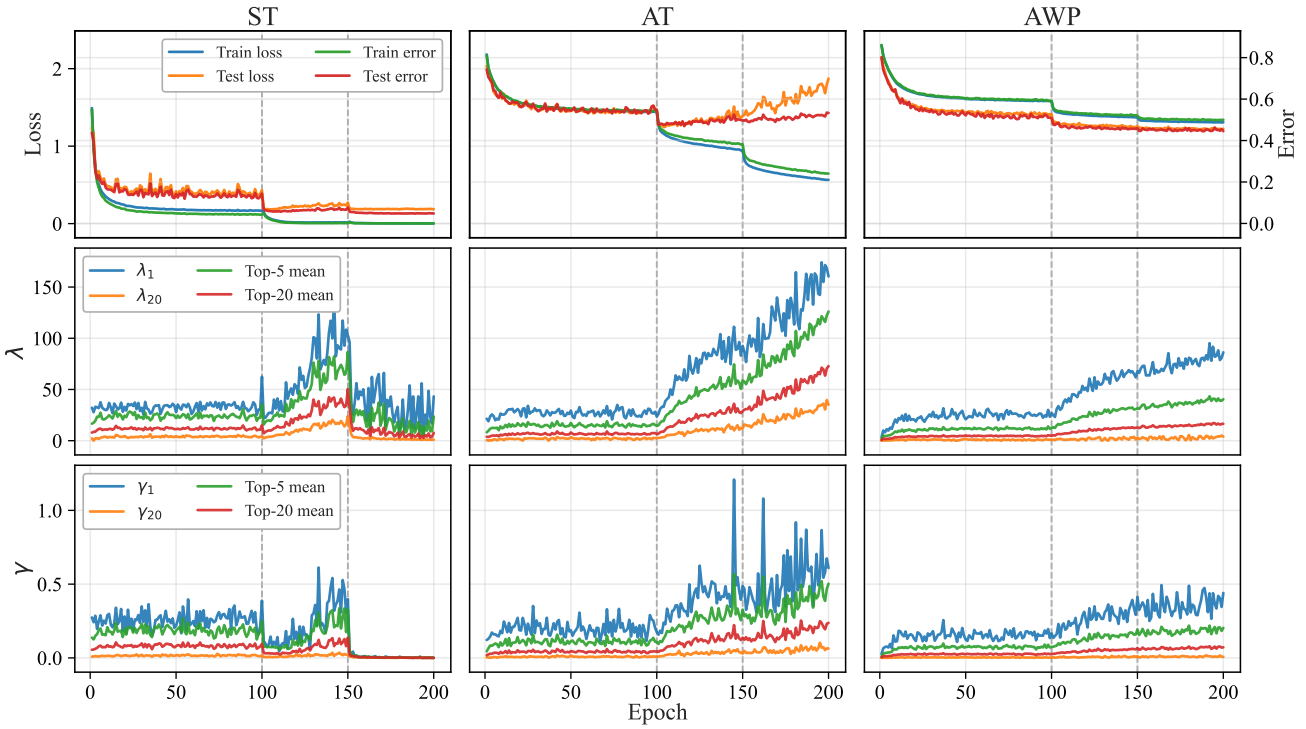

Figure 1: Learning curves across different learning algorithms on CIFAR-10: (first row) training/testing loss and error curves (ST: clean; AT/AWP: robust), (middle row) Hessian eigenvalues, and (last row) projected gradient noise variances.

5.2 PROGRESSIVE CURVATURE SHARPENING

Figure 1 visualizes the learning curves of statistics, including training and testing error/loss, top-k Hessian eigenvalues $\{\lambda_i\}_{i=1}^k$, and the corresponding projected gradient-noise variances $\{\gamma_i\}_{i=1}^k$, across different algorithms (ST, AT, and AWP). Note that the i-th pair (λ_i, γ_i) is associated with the i-th leading eigenvector of the Hessian matrix at the evaluated epoch. We follow the spectral estimation protocol described in Section 5.1 to obtain these estimates. To illustrate the dominating trend while keeping presentation cleanliness, we report the following summary spectral statistics: top-1, top-20, the mean of top-5, and the mean of top-20.

Notably, adversarial training depicts a stage-wise curve separated by the learning rate change. During the initial phase, both the robust loss and error steadily decrease until reaching a plateau, while the top Hessian eigenvalues and gradient noise stay small. After the learning rate decays by a factor of 10, both λ_i and γ_i exhibit a monotonically increasing trend, corresponding to a continual decrease in training robust error/loss but steadily worsening test statistics. The final stage is characterized by the smallest $\eta = 0.001$, the top Hessian eigenvalue remaining very large and continuing to increase, and the test robust accuracy dropping further. These observations align with the local stability condition in Section 4.2: top Hessian eigenvalues are capped by $2(1+\kappa)/\eta$, which is inversely proportional to the learning rate; a large learning rate η will restrict the optimization in low-curvature regions, while reducing η will relax the constraint, allowing the model to move into regions with higher curvature.

In sharp contrast to AT, the learning curves of ST demonstrate a different trend, especially in the final learning stage. Top Hessian eigenvalues are also progressively sharpened when $\eta = 0.01$, indicating the model is gradually fitting more fine-grained features within the training data; however, both λ_i and γ_i suddenly drop to close to zero and remain very low when η drops to 0.001. We note that the learning curves of Hessian eigenspectrum match the double descent phenomenon [Nakkiran et al., 2021], where SGD noise implicitly biases towards flat regions for over-parameterized neural networks. In addition, AWP penalizes the model's sharpness by perturbing weight parameters, leading to suppressed growth of both λ_i and γ_i in later training stages.

In our experiments, we observe that the top Hessian eigenvalues of adversarially trained models are much higher than those of standard trained models, especially when η is small. This suggests that optimizing adversarial training losses requires exploring high-curvature regions in the parameter space. The following proposition, proven in Appendix C.2, provides a theoretical justification for the above statement.

Proposition 5.1. *Optimizing the adversarial training loss $\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$ under L_2 or L_∞ norm to sufficiently low requires the top Hessian eigenvalues to be large, especially due to the increased cross-sample alignment of input and parameter gradients with respect to the robust feature subspace.*

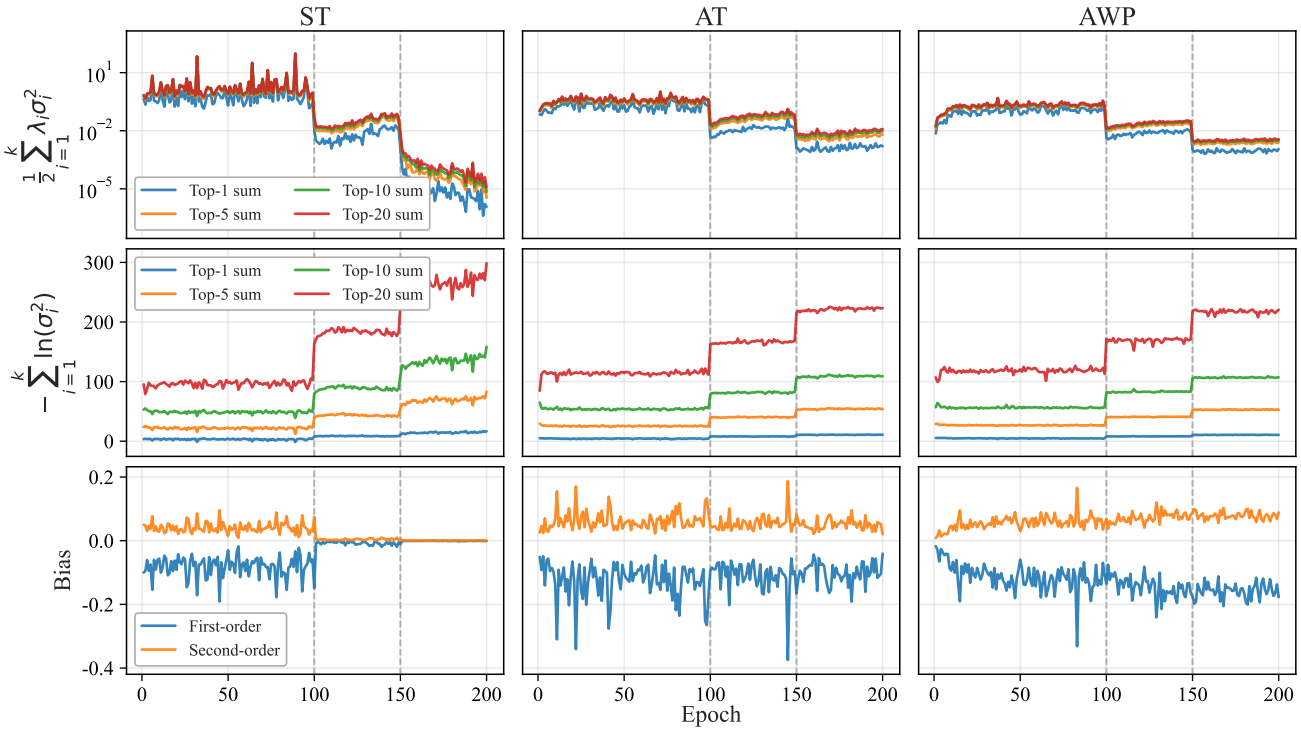

Figure 2: Learning curves of top-k bound diagnostics across different algorithms on CIFAR-10: (top row) curvature-weighted variance, (middle row) entropic KL, and (bottom row) first- and second-order bias terms.

Proposition 5.1 shows that to fit the training data well, robust learning algorithms need to explore high-curvature regions. Unlike ST, it is not possible for adversarial training to learn a model that is "flat" in all directions; a few large Hessian eigenvalues associated with robust features are necessary to ensure the input gradient sensitivity is sufficiently penalized.

5.3 POSTERIOR COLLAPSE & VARIANCE AMPLIFICATION

Based on the per-epoch spectral estimates, we can now empirically estimate the posterior mean, posterior covariance, and the bound decompositions in Theorem 4.6. The full bound contains KL terms beyond the log-determinant proxy, including the mean-norm term and the covariance-trace shift $\lambda_{t,i} \mapsto \lambda_{t,i} + 1/(\beta\sigma_{\mathcal{P}}^2)$ (see Appendix F.2 for full diagnostics). Specifically, we approximate the spectral estimates using top-k partial sums with $k \in \{1, 5, 10, 20\}$ for both the curvature-weighted variance term $\frac{1}{2}\sum_{i=1}^{k}\lambda_i\sigma_i^2$ and the entropic KL term $-\sum_{i=1}^{k}\ln(\sigma_i^2)$. We use the stationary estimates by default and switch to non-stationary recursion during the transient phase after η decays (Appendix D.2).

Figure 2 visualizes the learning curves across different algorithms. For AT, we observe that the inferred top-k posterior covariance contracts sharply after η decays at epochs 100 and 150, reflected by the stark increase in the entropic KL term and the sharp drop in the variance. This is evident if we derive the closed-form solutions for both terms under the stationary regime and assume commutativity (Lemma 4.4). Formally, for any $i = 1, 2, \ldots, k$, we have

$$\lambda_i\,\sigma_i^2 = \left(2 - \frac{\eta\lambda_i}{1+\kappa}\right)^{-1} \frac{\eta}{1-\kappa}\gamma_i,$$

$$\ln(\sigma_i^2) = \ln\left(\frac{\eta}{1-\kappa}\right) + \ln\left(\frac{\gamma_i}{\lambda_i}\right) - \ln\left(2 - \frac{\eta\lambda_i}{1+\kappa}\right).$$

Appendix F.1 provides empirical analysis examining the commutativity assumption, showing that the top Hessian eigendirections largely align with the gradient noise covariance. When η suddenly drops by a factor of 10, λ_i and γ_i remain similar, so the immediate change is primarily associated with the drop in η. Note that the decrease in $\lambda_i\,\sigma_i^2$ is approximately linear in η (dominating factor), whereas the rise in $-\ln(\sigma_i^2)$ is logarithmic in η (less drastic), which explains the initial drop of the testing curves in Figure 1.

As adversarial training continues, the curvature-weighted variance gradually increases, which coincides with increases in λ_i and γ_i. In comparison, the entropic KL term remains roughly constant during each stage. Note that, as illustrated in Figure 1, the top Hessian eigenvalues rise by a factor of 4 from epoch 100 to epoch 150 while noise variances increase by a factor of 2, jointly boosting the variance at the later stages of adversarial training. The observed trend aligns with the test-robust error curve during the onset of robust overfitting after the learning rate decay, suggesting that it has

a dominating influence on robust generalization. Besides, the first- and second-order biases remain at a consistent level throughout training, with only a few large spikes observed at certain epochs. We hypothesize that the spike is due to the (mis-)alignment of input gradients across minibatches.

Compared across the three tested learning algorithms, ST exhibits greater posterior contraction than AT, almost vanishing when $\eta = 0.001$. Such a collapsing posterior leads to a continuous decrease in curvature-weighted variance. The two bias terms approach zero after the learning rate decays. AWP, on the other hand, penalizes loss curvature, which benefits a more controlled variance and explains why it improves robust generalization. Interestingly, the two bias terms diverge gradually, suggesting that AWP may underfit the training objectives due to excessive penalization. These observations align with AWP's high training loss observed in Figure 1. Consequently, a promising future direction for our work is to develop a selective penalization scheme for AWP to reduce the training robust loss while preserving its generalization benefits. Combining with the insights from Proposition E.1, future designs should balance controlling the curvature-weighted variance to avoid robust overfitting while relaxing regularization strength in directions that capture robust features to enhance the fit of training data.

5.4 MECHANISTIC INTERPRETATION OF ROBUST OVERFITTING

Based on our time-resolved PAC-Bayesian robust generalization bounds and the corresponding empirical diagnostic curves, we can now summarize the underlying mechanisms of robust overfitting. Prior to the first learning-rate drop in adversarial training, the system reaches a stationary state. Due to the stability condition, Hessian eigenvalues are regularized: $0 < \lambda_i < 2(1+\kappa)/\eta$. However, as illustrated by Proposition 5.1, lowering adversarial loss requires stabilizing input sensitivity, suggesting the need to explore high-curvature regions to further improve optimization.

After η sharply decays, the system begins to explore high-curvature directions to further reduce the adversarial training loss. Since the system has only evolved gradually at the initial steps, the Hessian and gradient noise covariance remain similar; however, the posterior will rapidly concentrate due to the sharp learning-rate drop ($\eta \to \eta/10$), effectively lowering the *curvature-weighted variance* $\frac{1}{2}\sum_{i=1}^{m} \lambda_{t,i}\, \sigma_{t,i}^2$. While contracted posterior fluctuation also increases the *entropic KL-penalty* term $-\frac{1}{2\beta}\sum_{i=1}^{m} \ln \sigma_{t,i}^2$, it cannot offset the linear decrease in variance. Continual training with a small η keeps increasing Hessian eigenvalues, eventually boosting the variance term and worsening robust generalization. The above mechanistic interpretation matches the empirical observation of robust overfitting well: test robust error first quickly drops after a learning rate decay, then continues to rise, whereas the training loss keeps decreasing.

6 CONCLUSION

We derived time-resolved PAC-Bayesian robust generalization bounds by modeling momentum SGD as a discrete-time dynamical system and deriving closed-form posterior evolutions that depend on the learning rate, the loss geometry, and the stochastic gradient noise. Through theoretical analysis, empirical diagnostics, and controlled experiments, our work provides mechanistic insights into how learning dynamics lead to robust overfitting or generalization. Looking forward, important directions include extending our framework to more general settings, such as adaptive optimizers and relaxed assumptions, and leveraging the insights to enhance the training-time fit of AWP while avoiding overfitting.

AVAILABILITY

The code for implementing our spectral estimation algorithm and reproducing our main experimental results is available at: https://github.com/TrustMLRG/RobustGen.

References

Pierre Alquier et al. User-friendly introduction to PAC-Bayes bounds. *Foundations and Trends® in Machine Learning*, 17(2):174–303, 2024.

Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta numerica*, 30: 87–201, 2021.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

Ioar Casado, Luis A Ortega, Aritz Pérez, and Andrés R Masegosa. PAC-Bayes-Chernoff bounds for unbounded losses. *Advances in Neural Information Processing Systems*, 37:24350–24374, 2024.

Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Institute of Mathematical Statistics Lecture Notes–Monograph Series*. Institute of Mathematical Statistics, Beachwood, OH, 2007. ISBN 978-0-940600-72-0. doi: 10.1214/074921707000000391.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. BackPACK: Packing more into backprop. In *International Conference on Learning Representations*, 2020.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Shaopeng Fu and Di Wang. Theoretical analysis of robust overfitting for wide DNNs: An NTK approach. *IEEE Transactions on Information Theory*, 71(8):6311–6339, 2025. doi: 10.1109/TIT.2025.3570532.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

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

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 2021.

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

Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from ImageNet. `https://github.com/fastai/imagenette`, March 2019. GitHub repository.

Stanislaw Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019.

Gaojie Jin, Xinping Yi, Wei Huang, Sven Schewe, and Xiaowei Huang. Enhancing adversarial training with second-order statistics of weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Hoki Kim, Jinseong Park, Yujin Choi, and Jaewook Lee. Fantastic robustness measures: The secrets of robust generalization. *Advances in Neural Information Processing Systems*, 36:48793–48818, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Qianxiao Li, Cheng Tai, et al. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, 2017.

Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems*, 33:21476–21487, 2020.

Chen Liu, Zhichao Huang, Mathieu Salzmann, Tong Zhang, and Sabine Süsstrunk. On the impact of hard adversarial instances on overfitting in adversarial training. *Journal of Machine Learning Research*, 25(356):1–46, 2024.

Kangqiao Liu, Liu Ziyin, and Masahito Ueda. Noise and fluctuation of finite learning rate stochastic gradient descent. In *International Conference on Machine Learning*, 2021.

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*, 2018.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 2017.

David A McAllester. PAC-Bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, 1999.

Waleed Mustafa, Philipp Liznerski, Dennis Wagner, Puyu Wang, and M. Kloft. Non-vacuous PAC-Bayes bounds for models under adversarial corruptions. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

Waleed Mustafa, Philipp Liznerski, Antoine Ledent, Dennis Wagner, Puyu Wang, and Marius Kloft. Non-vacuous generalization bounds for adversarial risk in stochastic neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021 (12):124003, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, 2020.

R. Tyrrell Rockafellar and Roger J. B. Wets. *Variational Analysis*. Springer Berlin Heidelberg, 1998. ISBN 9783642024313. doi: 10.1007/978-3-642-02431-3.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, 2013.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Runzhi Tian and Yongyi Mao. Algorithmic stability based generalization bounds for adversarial training. In *The Thirteenth International Conference on Learning Representations*, 2025.

Nils Philipp Walter, Linara Adilova, Jilles Vreeken, and Michael Kamp. When flatness does (not) guarantee adversarial robustness. In *International Conference on Learning Representations*, 2026.

Zifan Wang, Nan Ding, Tomer Levinboim, Xi Chen, and Radu Soricut. Improving robust generalization by direct PAC-Bayesian bound minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, 2018.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 2020.

Jiancong Xiao, Yanbo Fan, Ruoyu Sun, Jue Wang, and Zhi-Quan Luo. Stability analysis and generalization bounds of adversarial training. *Advances in Neural Information Processing Systems*, 35:15446–15459, 2022.

Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. PAC-Bayesian spectrally-normalized bounds for adversarially robust generalization. *Advances in Neural Information Processing Systems*, 36:36305–36323, 2023.

Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. *Advances in Neural Information Processing Systems*, 34:26523–26535, 2021.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. PyHessian: Neural networks through the lens of the Hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

Liu Ziyin, Kangqiao Liu, Takashi Mori, and Masahito Ueda. Strength of minibatch noise in SGD. In *International Conference on Learning Representations*, 2022.

Liu Ziyin, Mingze Wang, Hongchao Li, and Lei Wu. Parameter symmetry and noise equilibrium of stochastic gradient descent. *Advances in Neural Information Processing Systems*, 37:93874–93906, 2024.

Liu Ziyin, Isaac L. Chuang, Tomer Galanti, and Tomaso A Poggio. Formation of representations in neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

How Learning Dynamics Drive Adversarially Robust Generalization? (Supplementary Material)

Yuelin Xu[1] **Xiao Zhang**[1]

[1] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

A PROOFS OF THEORETICAL RESULTS IN SECTION 4.1

A.1 PROOF OF LEMMA 4.1

Proof of Lemma 4.1. According to the definition of KL divergence, we have

$$\mathrm{KL}(\mathcal{Q}\,\|\,\mathcal{P}) = \int q(\boldsymbol{w})\,\ln\frac{q(\boldsymbol{w})}{p(\boldsymbol{w})}\,\mathrm{d}\boldsymbol{w} = \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\ln q(\boldsymbol{w}) - \ln p(\boldsymbol{w})]\,, \tag{18}$$

where the density functions of $\mathcal{P}$ and $\mathcal{Q}$ under our assumptions can be written as:

$$\begin{aligned} p(\boldsymbol{w}) &= (2\pi)^{-\frac{m}{2}}\,\sigma_{\mathcal{P}}^{-m}\cdot\exp\!\left(-\frac{1}{2\sigma_{\mathcal{P}}^2}\,\boldsymbol{w}^\top\boldsymbol{w}\right),\\ q(\boldsymbol{w}) &= (2\pi)^{-\frac{m}{2}}\,\det(\boldsymbol{\Sigma}_{\mathcal{Q}})^{-\frac{1}{2}}\cdot\exp\!\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})^\top\boldsymbol{\Sigma}_{\mathcal{Q}}^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})\right). \end{aligned}$$

Taking logs of the above equations, we obtain

$$\begin{aligned} \ln p(\boldsymbol{w}) &= -\frac{m}{2}\ln(2\pi) - m\ln\sigma_{\mathcal{P}} - \frac{1}{2\sigma_{\mathcal{P}}^2}\,\boldsymbol{w}^\top\boldsymbol{w},\\ \ln q(\boldsymbol{w}) &= -\frac{m}{2}\ln(2\pi) - \frac{1}{2}\ln\det(\boldsymbol{\Sigma}_{\mathcal{Q}}) - \frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})^\top\boldsymbol{\Sigma}_{\mathcal{Q}}^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}}). \end{aligned} \tag{19}$$

Plugging Equation 19 into Equation 18, we further obtain

$$\begin{aligned} \mathrm{KL}(\mathcal{Q}\,\|\,\mathcal{P}) &= \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[-\frac{1}{2}\ln\det(\boldsymbol{\Sigma}_{\mathcal{Q}}) - \frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})^\top\boldsymbol{\Sigma}_{\mathcal{Q}}^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}}) + m\ln\sigma_{\mathcal{P}} + \frac{1}{2\sigma_{\mathcal{P}}^2}\,\boldsymbol{w}^\top\boldsymbol{w}\right]\\ &= -\frac{1}{2}\ln\det(\boldsymbol{\Sigma}_{\mathcal{Q}}) - \frac{m}{2} + m\ln\sigma_{\mathcal{P}} + \frac{1}{2\sigma_{\mathcal{P}}^2}\left(\mathrm{Tr}(\boldsymbol{\Sigma}_{\mathcal{Q}}) + \|\boldsymbol{\mu}_{\mathcal{Q}}\|_2^2\right), \end{aligned}$$

where the last equality holds because $\mathcal{Q} = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\Sigma}_{\mathcal{Q}})$. Specifically, for any $\boldsymbol{w}\sim\mathcal{Q}$, $\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, which suggests that $\mathbb{E}_{\mathcal{Q}}\left[(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})^\top\boldsymbol{\Sigma}_{\mathcal{Q}}^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}})\right] = m$, and $\mathbb{E}_{\mathcal{Q}}\left[\boldsymbol{w}^\top\boldsymbol{w}\right] = \mathbb{E}_{\mathcal{Q}}\left[\|\boldsymbol{w}-\boldsymbol{\mu}_{\mathcal{Q}}\|_2^2\right] + \|\boldsymbol{\mu}_{\mathcal{Q}}\|_2^2 = \mathrm{Tr}(\boldsymbol{\Sigma}_{\mathcal{Q}}) + \|\boldsymbol{\mu}_{\mathcal{Q}}\|_2^2$. Therefore, we complete the proof of Lemma 4.1. □

A.2 PROOF OF LEMMA 4.2

Proof of Lemma 4.2. Under the local quadratic upper-bound assumption in Lemma 4.2, the empirical adversarial loss satisfies the following Taylor upper expansion around the reference point $\tilde{\boldsymbol{w}}$:

$$\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}) \le \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}})^\top(\boldsymbol{w}-\tilde{\boldsymbol{w}}) + \frac{1}{2}\,(\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})(\boldsymbol{w}-\tilde{\boldsymbol{w}}) + r_{\tilde{\boldsymbol{w}}}(\boldsymbol{w}),\ \text{for any } \boldsymbol{w}\sim\mathcal{Q}, \tag{20}$$

where $\nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}})$ (resp. $\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})$) denotes the gradient (resp. Hessian) of the empirical adversarial loss at the reference point. Taking expectation over $\boldsymbol{w} \sim \mathcal{Q} = \mathcal{N}(\boldsymbol{\mu}_\mathcal{Q}, \boldsymbol{\Sigma}_\mathcal{Q})$, we have

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\Big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\Big] \le \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}})^\top \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[(\boldsymbol{w}-\tilde{\boldsymbol{w}})] + \frac{1}{2}\,\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\Big[(\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})(\boldsymbol{w}-\tilde{\boldsymbol{w}})\Big] + \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[r_{\tilde{\boldsymbol{w}}}(\boldsymbol{w})]. \tag{21}$$

Let $\Delta\bar{\boldsymbol{w}} := \boldsymbol{\mu}_\mathcal{Q} - \tilde{\boldsymbol{w}}$ denote the posterior mean shift. Since $\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\boldsymbol{w}] = \boldsymbol{\mu}_\mathcal{Q}$, we have

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[(\boldsymbol{w}-\tilde{\boldsymbol{w}})] = \boldsymbol{\mu}_\mathcal{Q} - \tilde{\boldsymbol{w}} = \Delta\bar{\boldsymbol{w}}. \tag{22}$$

The remaining task is to simplify the quadratic term in Equation 21. From the second-moment identity, we know

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[(\boldsymbol{w}-\tilde{\boldsymbol{w}})(\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top\big] = \boldsymbol{\Sigma}_\mathcal{Q} + \Delta\bar{\boldsymbol{w}}\,\Delta\bar{\boldsymbol{w}}^\top, \tag{23}$$

which holds for any Gaussian with mean $\boldsymbol{\mu}_\mathcal{Q}$ and covariance $\boldsymbol{\Sigma}_\mathcal{Q}$. Hence, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\Big[(\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})(\boldsymbol{w}-\tilde{\boldsymbol{w}})\Big] &= \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[\mathrm{Tr}\Big((\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})(\boldsymbol{w}-\tilde{\boldsymbol{w}})\Big)\right] \\
&= \mathrm{Tr}\Big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\,\mathbb{E}\big[(\boldsymbol{w}-\tilde{\boldsymbol{w}})(\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top\big]\Big) \\
&= \mathrm{Tr}\Big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\boldsymbol{\Sigma}_\mathcal{Q}\Big) + \mathrm{Tr}\Big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\Delta\bar{\boldsymbol{w}}\,\Delta\bar{\boldsymbol{w}}^\top\Big) \\
&= \mathrm{Tr}\Big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\boldsymbol{\Sigma}_\mathcal{Q}\Big) + \Delta\bar{\boldsymbol{w}}^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\Delta\bar{\boldsymbol{w}},
\end{aligned} \tag{24}$$

where the second and the last equalities hold because of the cyclic property of trace: $\mathrm{Tr}(\mathbf{A}\mathbf{u}\mathbf{u}^\top) = \mathrm{Tr}(\mathbf{u}^\top\mathbf{A}\mathbf{u})$ holds for any matrix $\mathbf{A}$ and vector $\boldsymbol{u}$, and the third equality is due to Equation 23.

Plugging Equation 22 and Equation 24 into Equation 21 yields

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] \le \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}})^\top\Delta\bar{\boldsymbol{w}} + \frac{1}{2}\Delta\bar{\boldsymbol{w}}^\top\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\Delta\bar{\boldsymbol{w}} + \frac{1}{2}\mathrm{Tr}(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\boldsymbol{\Sigma}_\mathcal{Q}) + R_\mathcal{Q}(\tilde{\boldsymbol{w}}),$$

where $R_\mathcal{Q}(\tilde{\boldsymbol{w}}) = \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[r_{\tilde{\boldsymbol{w}}}(\boldsymbol{w})]$. This completes the proof of Lemma 4.2. □

A.3 PROOF OF THEOREM 4.3

Proof of Theorem 4.3. We start from the generic PAC-Bayesian robust generalization bound in Theorem 3.2:

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\mathcal{L}_\epsilon(\boldsymbol{w})\big] \le \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] + \frac{1}{\beta}\mathrm{KL}(\mathcal{Q}\,\|\,\mathcal{P}) + \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta}\ln\alpha. \tag{25}$$

Step 1: Expected empirical adversarial loss under a Gaussian posterior. Under the local quadratic upper bound of Lemma 4.2, we have

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] \le \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}) + \big\langle\nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}), \Delta\bar{\boldsymbol{w}}\big\rangle + \frac{1}{2}\Delta\bar{\boldsymbol{w}}^\top\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\Delta\bar{\boldsymbol{w}} + \frac{1}{2}\mathrm{Tr}\big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})\boldsymbol{\Sigma}_\mathcal{Q}\big) + R_\mathcal{Q}(\tilde{\boldsymbol{w}}), \tag{26}$$

where $\Delta\bar{\boldsymbol{w}} = \boldsymbol{\mu}_\mathcal{Q} - \tilde{\boldsymbol{w}}$.

Step 2: Closed-form KL divergence. For the spherical Gaussian prior $\mathcal{P} = \mathcal{N}(\mathbf{0}, \sigma_\mathcal{P}^2\mathbf{I})$ and Gaussian posterior $\mathcal{Q} = \mathcal{N}(\boldsymbol{\mu}_\mathcal{Q}, \boldsymbol{\Sigma}_\mathcal{Q})$, Lemma 4.1 yields

$$\frac{1}{\beta}\mathrm{KL}(\mathcal{Q}\,\|\,\mathcal{P}) = \underbrace{\frac{1}{2\beta\sigma_\mathcal{P}^2}\Big(\|\boldsymbol{\mu}_\mathcal{Q}\|_2^2 + \mathrm{Tr}(\boldsymbol{\Sigma}_\mathcal{Q})\Big) - \frac{1}{2\beta}\ln\det(\boldsymbol{\Sigma}_\mathcal{Q})}_{\text{terms depending on } \mathcal{Q}} + \underbrace{\frac{m}{2\beta}\ln(\sigma_\mathcal{P}^2) - \frac{m}{2\beta}}_{\text{constant in } \mathcal{Q}}. \tag{27}$$

Step 3: Combine terms. Substituting equation 26 and equation 27 into equation 25, and grouping all terms independent of the posterior $\mathcal{Q}$ into

$$C_{\mathrm{PB}} = C_{\mathrm{PB}}(\sigma_\mathcal{P}^2, \mathcal{S}, \alpha, \beta) = \frac{m}{2\beta}\ln\sigma_\mathcal{P}^2 - \frac{m}{2\beta} + \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta}\ln\alpha, \tag{28}$$

where we obtain exactly the bound in Equation 10, including the local Taylor upper remainder. □

A.4 ROBUST GENERALIZATION BOUNDS UNDER GAUSSIAN-MIXTURE POSTERIOR

Corollary A.1 (Robust Generalization with Gaussian Mixtures)**.** *Let $\mathcal{P} = \mathcal{N}(0, \sigma_{\mathcal{P}}^2 \mathbf{I})$ be the prior, and let the posterior be a mixture of Gaussians:*

$$\mathcal{Q} = \sum_{\ell=1}^{L} \pi_\ell \, \mathcal{Q}_\ell, \quad \textit{where } \mathcal{Q}_\ell = \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell), \; \sum_{\ell=1}^{L} \pi_\ell = 1, \; \textit{and } \pi_\ell \geq 0. \tag{29}$$

Assume the generalized local quadratic upper bound (Lemma 4.2) holds for each component ℓ around a reference point $\boldsymbol{w}_{ref,\ell}$ with Hessian $\mathbf{H}_\ell$, gradient $\boldsymbol{g}_\ell := \nabla \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{ref,\ell})$, and remainder $R_{\mathcal{Q}_\ell}(\boldsymbol{w}_{ref,\ell})$. For any $\beta > 0$ and $\alpha \in (0,1)$, with probability at least $1-\alpha$, we have

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\mathcal{L}_\epsilon(\boldsymbol{w})] \;\leq\; \sum_{\ell=1}^{L} \pi_\ell \Big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{ref,\ell}) + \underbrace{\langle \boldsymbol{g}_\ell, \, \boldsymbol{\mu}_\ell - \boldsymbol{w}_{ref,\ell} \rangle}_{\textit{first-order bias}} + \underbrace{\frac{1}{2}(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{ref,\ell})^\top \mathbf{H}_\ell (\boldsymbol{\mu}_\ell - \boldsymbol{w}_{ref,\ell})}_{\textit{second-order bias}} + \underbrace{\frac{1}{2}\mathrm{Tr}(\mathbf{H}_\ell \boldsymbol{\Sigma}_\ell)}_{\textit{variance}} + R_{\mathcal{Q}_\ell}(\boldsymbol{w}_{ref,\ell})\Big]$$
$$+ \sum_{\ell=1}^{L} \frac{\pi_\ell}{2\beta} \left(\frac{\mathrm{Tr}(\boldsymbol{\Sigma}_\ell)}{\sigma_{\mathcal{P}}^2} + \frac{\|\boldsymbol{\mu}_\ell\|_2^2}{\sigma_{\mathcal{P}}^2} - m + m \ln \sigma_{\mathcal{P}}^2 - \ln \det \boldsymbol{\Sigma}_\ell \right) + \; \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta} \ln \alpha. \tag{30}$$

Here C_ℓ denotes the global scalar loss-bound constant, not the mixture-component index. When $\mathbf{H}_\ell \succ 0$, the sum of the first- and second-order bias terms can be written by completing the square, and is minimized at the Newton-corrected mean $\boldsymbol{\mu}_\ell = \boldsymbol{w}_{ref,\ell} - \mathbf{H}_\ell^{-1} \boldsymbol{g}_\ell$, recovering the familiar optimization gain $-\frac{1}{2}\boldsymbol{g}_\ell^\top \mathbf{H}_\ell^{-1} \boldsymbol{g}_\ell$.

Proof of Corollary A.1. We start with the most general PAC-Bayesian bound in Theorem 3.2, since it holds for any posterior $\mathcal{Q}$. For any $\beta > 0$ and any $\alpha \in (0,1)$, with probability at least $1-\alpha$ over the finite sample set $\mathcal{S}$, we have

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\mathcal{L}_\epsilon(\boldsymbol{w})] \leq \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[\frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},y)\in\mathcal{S}} \ell_{\mathrm{adv}}(\boldsymbol{w}, \boldsymbol{x}, y)\right] + \frac{1}{\beta}\mathrm{KL}(\mathcal{Q} \,\|\, \mathcal{P}) + \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta}\ln\alpha. \tag{31}$$

We now instantiate this bound by choosing the posterior to be a mixture of Gaussian distributions specified in Equation 29 and the prior $\mathcal{P} = \mathcal{N}(0, \sigma_{\mathcal{P}}^2 \mathbf{I})$.

Step 1: Decomposition of the empirical adversarial loss. By the definition of expectations under mixture distributions, for any measurable function $f : \mathcal{W} \to \mathbb{R}$,

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[f(\boldsymbol{w})] = \sum_{\ell=1}^{L} \pi_\ell \, \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}_\ell}[f(\boldsymbol{w})]. \tag{32}$$

Applying Equation 32 to $f(\boldsymbol{w}) = \hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$ gives

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] = \sum_{\ell=1}^{L} \pi_\ell \, \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}_\ell}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big]. \tag{33}$$

For each Gaussian component $\mathcal{Q}_\ell$, note that we assume $\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$ can be locally upper bounded around the basin-specific reference point $\boldsymbol{w}_{\mathrm{ref},\ell}$ by a quadratic form with Hessian $\mathbf{H}_\ell$ and gradient $\boldsymbol{g}_\ell := \nabla\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\mathrm{ref},\ell})$. Thus, applying Lemma 4.2 to each component, we obtain

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}_\ell}\big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\big] \leq \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\mathrm{ref},\ell}) + \boldsymbol{g}_\ell^\top(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell}) + \frac{1}{2}(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell})^\top \mathbf{H}_\ell (\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell}) + \frac{1}{2}\mathrm{Tr}(\mathbf{H}_\ell\boldsymbol{\Sigma}_\ell) + R_{\mathcal{Q}_\ell}(\boldsymbol{w}_{\mathrm{ref},\ell}). \tag{34}$$

Substituting Equation 34 into Equation 33 yields the mixture-expanded empirical loss:

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})] \leq \sum_{\ell=1}^{L} \pi_\ell \Big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\mathrm{ref},\ell}) + \boldsymbol{g}_\ell^\top(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell})$$
$$+ \frac{1}{2}(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell})^\top \mathbf{H}_\ell (\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell}) + \frac{1}{2}\mathrm{Tr}(\mathbf{H}_\ell\boldsymbol{\Sigma}_\ell) + R_{\mathcal{Q}_\ell}(\boldsymbol{w}_{\mathrm{ref},\ell})\Big]. \tag{35}$$

Step 2: KL divergence upper bound for the mixture posterior. Since KL divergence is convex in its first argument, the mixture posterior $\mathcal{Q} = \sum_{\ell=1}^{L} \pi_\ell \mathcal{Q}_\ell$ satisfies

$$\mathrm{KL}(\mathcal{Q} \,\|\, \mathcal{P}) = \mathrm{KL}\Big(\sum_{\ell=1}^{L} \pi_\ell \mathcal{Q}_\ell \,\Big\|\, \mathcal{P}\Big) \le \sum_{\ell=1}^{L} \pi_\ell \,\mathrm{KL}(\mathcal{Q}_\ell \,\|\, \mathcal{P}), \tag{36}$$

where the inequality follows from the definition KL divergence and the log sum inequality. For each $\mathcal{Q}_\ell = \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$, Lemma 4.1 provides the closed-form expression:

$$\mathrm{KL}(\mathcal{Q}_\ell \,\|\, \mathcal{P}) = \frac{\mathrm{Tr}(\boldsymbol{\Sigma}_\ell)}{2\sigma_{\mathcal{P}}^2} + \frac{\|\boldsymbol{\mu}_\ell\|_2^2}{2\sigma_{\mathcal{P}}^2} - \frac{m}{2} + \frac{m}{2}\ln\sigma_{\mathcal{P}}^2 - \frac{1}{2}\ln\det\boldsymbol{\Sigma}_\ell. \tag{37}$$

Multiplying Equation 36 by $1/\beta$ and substituting Equation 37 yields

$$\frac{1}{\beta}\mathrm{KL}(\mathcal{Q} \,\|\, \mathcal{P}) \le \sum_{\ell=1}^{L} \frac{\pi_\ell}{2\beta}\left(\frac{\mathrm{Tr}(\boldsymbol{\Sigma}_\ell)}{\sigma_{\mathcal{P}}^2} + \frac{\|\boldsymbol{\mu}_\ell\|_2^2}{\sigma_{\mathcal{P}}^2} - m + m\ln\sigma_{\mathcal{P}}^2 - \ln\det\boldsymbol{\Sigma}_\ell\right). \tag{38}$$

Finally, we substitute the empirical-loss expansion (Equation 35) and the KL bound (Equation 38) into Equation 31. Noticing that the remaining terms, where C_ℓ denotes the global scalar loss-bound constant, $\frac{\beta C_\ell^2}{8|\mathcal{S}|}$ and $-\frac{1}{\beta}\ln(\alpha)$ do not depend on the mixture-component index ℓ, we obtain

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}[\mathcal{L}_\epsilon(\boldsymbol{w})] \le \sum_{\ell=1}^{L} \pi_\ell\Big[\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\mathrm{ref},\ell}) + \boldsymbol{g}_\ell^\top(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell}) + \frac{1}{2}(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell})^\top\mathbf{H}_\ell(\boldsymbol{\mu}_\ell - \boldsymbol{w}_{\mathrm{ref},\ell}) + \frac{1}{2}\mathrm{Tr}(\mathbf{H}_\ell\boldsymbol{\Sigma}_\ell) + R_{\mathcal{Q}_\ell}(\boldsymbol{w}_{\mathrm{ref},\ell})\Big]$$

$$+ \sum_{\ell=1}^{L} \frac{\pi_\ell}{2\beta}\left(\frac{\mathrm{Tr}(\boldsymbol{\Sigma}_\ell)}{\sigma_{\mathcal{P}}^2} + \frac{\|\boldsymbol{\mu}_\ell\|_2^2}{\sigma_{\mathcal{P}}^2} - m + m\ln\sigma_{\mathcal{P}}^2 - \ln\det\boldsymbol{\Sigma}_\ell\right) + \frac{\beta C_\ell^2}{8|\mathcal{S}|} - \frac{1}{\beta}\ln\alpha.$$

This expression gives the stated upper bound in Corollary A.1, completing the proof. □

B PROOFS OF THEORETICAL RESULTS IN SECTION 4.2

B.1 STATIONARY REGIME: PROOF OF LEMMA 4.4

Proof of Lemma 4.4. Under the generalized quadratic loss approximation, the stochastic gradient includes a constant drift $\boldsymbol{g}_{\mathrm{ref}} := \nabla\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\mathrm{ref}})$. For covariance calculations, we can omit this term without loss of generality, because covariance measures *centered* fluctuations: letting $\tilde{\boldsymbol{u}}_t = \boldsymbol{u}_t - \mathbb{E}[\boldsymbol{u}_t]$ denote the centered state, the drift appears identically in both $\boldsymbol{u}_t$ and $\mathbb{E}[\boldsymbol{u}_t]$ and cancels upon subtraction. This cancellation holds regardless of whether the system is stationary or transient (the multi-step transient recursion is handled in the proof of Lemma 4.5).

Throughout this proof, we treat the training set $\mathcal{S}$ (and hence the empirical loss $\hat{\mathcal{L}}_\epsilon$) as fixed, and take all expectations/covariances with respect to the algorithmic randomness conditional on $\mathcal{S}$.

State-space form. To begin with, we prove Equation 12. From the updates in Equation 4, we have

$$\begin{aligned}\boldsymbol{v}_{t+1} &= \kappa\,\boldsymbol{v}_t + \boldsymbol{g}_{\mathrm{ref}} + \mathbf{H}_{\mathrm{ref}}(\boldsymbol{w}_t - \boldsymbol{w}_{\mathrm{ref}}) + \boldsymbol{\xi}_t,\\ \boldsymbol{w}_{t+1} - \boldsymbol{w}_{\mathrm{ref}} &= (\boldsymbol{w}_t - \boldsymbol{w}_{\mathrm{ref}}) - \eta\boldsymbol{v}_{t+1} = (\mathbf{I} - \eta\mathbf{H}_{\mathrm{ref}})(\boldsymbol{w}_t - \boldsymbol{w}_{\mathrm{ref}}) - \eta\kappa\,\boldsymbol{v}_t - \eta\,\boldsymbol{g}_{\mathrm{ref}} - \eta\,\boldsymbol{\xi}_t.\end{aligned}$$

Stacking the two lines with the joint state $\boldsymbol{u}_t := \begin{bmatrix}\boldsymbol{w}_t - \boldsymbol{w}_{\mathrm{ref}}\\ \boldsymbol{v}_t\end{bmatrix} \in \mathbb{R}^{2m}$, we obtain the affine state-space system

$$\boldsymbol{u}_{t+1} \;=\; \mathbf{A}\,\boldsymbol{u}_t + \mathbf{G}\left[\boldsymbol{g}_{\mathrm{ref}} + \boldsymbol{\xi}_t\right], \quad \mathbf{A} = \begin{bmatrix}\mathbf{I} - \eta\mathbf{H}_{\mathrm{ref}} & -\eta\kappa\,\mathbf{I}\\ \mathbf{H}_{\mathrm{ref}} & \kappa\,\mathbf{I}\end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix}-\eta\mathbf{I}\\ \mathbf{I}\end{bmatrix},$$

which is exactly equation 12. □

Stationary Mean & Covariance. Next, we derive the stationary posterior mean and covariance as in Equations 13 and 14.

Remark B.1 (Commutative Structure and Spectral Decomposition). When $\mathbf{C}$ commutes with $\mathbf{H}_{\text{ref}}$, they share the same eigenbasis. Let $\{\lambda_1, \ldots, \lambda_m\}$ and $\{\gamma_1, \ldots, \gamma_m\}$ denote the eigenvalues of $\mathbf{H}_{\text{ref}}$ and $\mathbf{C}$, respectively. As shown in the derivation of Equation 14, the eigenvalues of the stationary covariance $\mathbf{\Sigma}$ are:

$$\forall i \in \{1, \ldots, m\}, \quad \sigma_i^2 = \frac{\eta}{1-\kappa} \cdot \frac{\gamma_i}{\lambda_i \left(2 - \frac{\eta}{1+\kappa}\lambda_i\right)}. \tag{39}$$

The stability condition requires $0 < \lambda_i < \frac{2(1+\kappa)}{\eta}$ for all i. Since both $\mathbf{H}_{\text{ref}}$ and $\mathbf{C}$ depend on the perturbation strength ϵ, altering ϵ correspondingly influences the stationary covariance structure.

Derivation of Equation 13. Under the local quadratic approximation in Lemma 4.2, the gradient of the empirical adversarial loss at any point $\boldsymbol{w}$ can be written as:

$$\nabla_{\boldsymbol{w}} \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}) = \nabla \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\text{ref}}) + \mathbf{H}_{\text{ref}}(\boldsymbol{w} - \boldsymbol{w}_{\text{ref}}),$$

where $\nabla \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\text{ref}}) =: \boldsymbol{g}_{\text{ref}}$ is the (possibly non-zero) gradient at the reference point $\boldsymbol{w}_{\text{ref}}$.

Taking expectation over the posterior $\mathcal{Q} = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{Q}}, \mathbf{\Sigma}_{\mathcal{Q}})$, we obtain

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[\nabla_{\boldsymbol{w}} \hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\right] = \boldsymbol{g}_{\text{ref}} + \mathbf{H}_{\text{ref}}(\boldsymbol{\mu}_{\mathcal{Q}} - \boldsymbol{w}_{\text{ref}}). \tag{40}$$

Since the posterior $\mathcal{Q}$ is stationary and obtained via SGD on $\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$, the expected gradient must vanish at stationarity. Otherwise, running another step of SGD would shift the mean and break the stationary assumption. Thus,

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[\nabla_{\boldsymbol{w}} \hat{\mathcal{L}}_\epsilon(\boldsymbol{w})\right] = \mathbf{0}.$$

Combining with Equation 40, we have

$$\boldsymbol{g}_{\text{ref}} + \mathbf{H}_{\text{ref}}(\boldsymbol{\mu}_{\mathcal{Q}} - \boldsymbol{w}_{\text{ref}}) = \mathbf{0}.$$

By the positive-definiteness assumption in Lemma 4.4, $\mathbf{H}_{\text{ref}} \succ 0$, and hence $\mathbf{H}_{\text{ref}}$ is invertible. Solving for $\boldsymbol{\mu}_{\mathcal{Q}}$ yields

$$\boldsymbol{\mu}_{\mathcal{Q}} = \boldsymbol{w}_{\text{ref}} - \mathbf{H}_{\text{ref}}^{-1} \nabla \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}_{\text{ref}}),$$

which matches Equation 13 after identifying $\boldsymbol{w}_{\text{ref}}$ with $\tilde{\boldsymbol{w}}$ and $\mathbf{H}_{\text{ref}}$ with $\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})$. This stationary mean relation has the form of a Newton-type correction around the reference point. □

Derivation of Equation 14. *Joint Lyapunov equation and projection.* From Lemma 4.4, the stacked state $\boldsymbol{u}_t = \begin{bmatrix} \boldsymbol{w}_t - \boldsymbol{w}_{\text{ref}} \\ \boldsymbol{v}_t \end{bmatrix} \in \mathbb{R}^{2m}$ obeys the affine recursion

$$\boldsymbol{u}_{t+1} = \mathbf{A}\,\boldsymbol{u}_t + \mathbf{G}\big(\boldsymbol{g}_{\text{ref}} + \boldsymbol{\xi}_t\big), \qquad \mathbf{A} = \begin{bmatrix} \mathbf{I} - \eta\mathbf{H}_{\text{ref}} & -\eta\kappa\,\mathbf{I} \\ \mathbf{H}_{\text{ref}} & \kappa\,\mathbf{I} \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} -\eta\mathbf{I} \\ \mathbf{I} \end{bmatrix}.$$

Since covariance is translation-invariant, the drift $\boldsymbol{g}_{\text{ref}}$ does not affect it: defining the centered state $\tilde{\boldsymbol{u}}_t := \boldsymbol{u}_t - \mathbb{E}[\boldsymbol{u}_t \mid \mathcal{S}]$, we have $\tilde{\boldsymbol{u}}_{t+1} = \mathbf{A}\,\tilde{\boldsymbol{u}}_t + \mathbf{G}\,\boldsymbol{\xi}_t$. Assume the mini-batch noises $\{\boldsymbol{\xi}_t\}$ satisfy the martingale-difference (conditional unbiasedness) property $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathcal{F}_t, \mathcal{S}] = \mathbf{0}$, so that $\mathrm{Cov}(\tilde{\boldsymbol{u}}_t, \boldsymbol{\xi}_t \mid \mathcal{S}) = \mathbf{0}$. Then taking covariances at stationarity yields the discrete Lyapunov equation

$$\mathbf{\Sigma}_{\text{joint}} = \mathbf{A}\,\mathbf{\Sigma}_{\text{joint}}\,\mathbf{A}^\top + \mathbf{G}\,\mathbf{C}\,\mathbf{G}^\top, \qquad \mathbf{C} := \mathrm{Cov}(\boldsymbol{\xi}_t \mid \mathcal{S}).$$

Let $\mathbf{\Pi} = [\,\mathbf{I}\;\;\mathbf{0}\,] \in \mathbb{R}^{m\times 2m}$ be the projection onto the parameter component; then $\mathbf{\Sigma} = \mathrm{Cov}(\boldsymbol{w}_t) = \mathbf{\Pi}\,\mathbf{\Sigma}_{\text{joint}}\,\mathbf{\Pi}^\top$, which yields the projected Lyapunov characterization of the stationary parameter covariance.

Closed form under commutativity. Assume $\mathbf{H}_{\text{ref}} \succ 0$, $\mathbf{C}$ is finite, and $\mathbf{H}_{\text{ref}}\mathbf{C} = \mathbf{C}\mathbf{H}_{\text{ref}}$. Then there exists an orthogonal $\mathbf{U}$ such that $\mathbf{H}_{\text{ref}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, $\mathbf{C} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^\top$ with $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ and $\mathbf{\Gamma} = \mathrm{diag}(\gamma_1, \ldots, \gamma_m)$. Define the block-orthogonal transform $\mathbf{T} := \mathrm{diag}(\mathbf{U}, \mathbf{U})$ and rotated state/noise $\boldsymbol{z}_t := \mathbf{T}^\top \boldsymbol{u}_t$, $\boldsymbol{\zeta}_t := \mathbf{U}^\top \boldsymbol{\xi}_t$. In this eigenbasis, the dynamics decouple:

$$\boldsymbol{z}_t = \widetilde{\mathbf{A}}\,\boldsymbol{z}_{t-1} + \widetilde{\mathbf{G}}\,\boldsymbol{\zeta}_{t-1}, \text{ where } \widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{I} - \eta\mathbf{\Lambda} & -\eta\kappa\,\mathbf{I} \\ \mathbf{\Lambda} & \kappa\,\mathbf{I} \end{bmatrix}, \quad \widetilde{\mathbf{G}} = \begin{bmatrix} -\eta\mathbf{I} \\ \mathbf{I} \end{bmatrix}, \quad \mathrm{Cov}(\boldsymbol{\zeta}_t) = \mathbf{\Gamma}.$$

Hence each eigendirection i follows a 2×2 "heavy-ball" system with curvature $\lambda = \lambda_i$ and noise variance $\gamma = \gamma_i$:

$$\mathbf{A}(\lambda) = \begin{bmatrix} 1 - \eta\lambda & -\eta\kappa \\ \lambda & \kappa \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} -\eta \\ 1 \end{bmatrix}, \quad \mathbf{Q}(\gamma) = \mathbf{G}\,\gamma\,\mathbf{G}^\top = \begin{bmatrix} \gamma\eta^2 & -\gamma\eta \\ -\gamma\eta & \gamma \end{bmatrix}.$$

Let $\mathbf{S} = \begin{bmatrix} x & y \\ y & z \end{bmatrix}$ be the stationary joint covariance in this mode, solving $\mathbf{S} = \mathbf{A}(\lambda)\,\mathbf{S}\,\mathbf{A}(\lambda)^\top + \mathbf{Q}(\gamma)$. Solving the resulting linear system in x, y, z (unique under stability) gives the parameter variance

$$x = \frac{\gamma\,\eta(1+\kappa)}{\lambda(1-\kappa)\,\big(2(1+\kappa) - \eta\lambda\big)}.$$

Therefore, in the eigenbasis the stationary parameter covariance is diagonal with entries

$$\frac{\eta}{1-\kappa} \cdot \frac{\gamma_i}{\lambda_i\left(2 - \frac{\eta}{1+\kappa}\lambda_i\right)} \;=\; \Big(\lambda_i\big(2 - \frac{\eta}{1+\kappa}\lambda_i\big)\Big)^{-1} \frac{\eta}{1-\kappa}\,\gamma_i,$$

and conjugating back by $\mathbf{U}$ yields

$$\mathbf{\Sigma} \;=\; \left[\,\mathbf{H}_{\text{ref}}\left(2\mathbf{I} - \frac{\eta}{1+\kappa}\mathbf{H}_{\text{ref}}\right)\,\right]^{-1} \;\frac{\eta}{1-\kappa}\,\mathbf{C},$$

which matches Equation 14 after identifying $\mathbf{H}_{\text{ref}}$ with $\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}})$.

Stability condition. For the scalar mode, the characteristic polynomial of $\mathbf{A}(\lambda)$ is $t^2 - (1 - \eta\lambda + \kappa)t + \kappa$. All roots lie in the open unit disk iff $|\kappa| < 1$ and $0 < \eta\lambda < 2(1+\kappa)$ (e.g., Jury criterion). Since $\kappa \in [0, 1)$, this equivalently requires $0 < \frac{\eta}{1+\kappa}\lambda_i < 2$ for every i. Under this condition, the stationary covariance exists and is finite. □

B.2 NON-STATIONARY REGIME: PROOF OF LEMMA 4.5

Proof of Lemma 4.5. Fix a window start time t and let $t' = t + k$. Under the within-window invariance assumption, the linearized SGD dynamics (Lemma 4.4) take the affine state-space form

$$\boldsymbol{u}_{s+1} = \mathbf{A}_t\boldsymbol{u}_s + \mathbf{G}\big[\nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t) + \boldsymbol{\xi}_s\big], \qquad s = t, t+1, \ldots, t'-1,$$

where $\boldsymbol{u}_s = [\boldsymbol{w}_s - \tilde{\boldsymbol{w}}_t; \boldsymbol{v}_s]$ and $\mathbb{E}[\boldsymbol{\xi}_s \mid \mathcal{F}_s, \mathcal{S}] = \mathbf{0}$. The finite-noise assumption ensures that the fixed conditional covariance $\mathbf{C}_t = \mathbb{E}[\boldsymbol{\xi}_s\boldsymbol{\xi}_s^\top \mid \mathcal{F}_s, \mathcal{S}]$ exists in the window, and the mode-wise stability condition $0 < \eta_t\lambda_{t,i} < 2(1+\kappa)$ for the fixed learning rate used inside the window guarantees that the propagated covariance terms remain finite in the analyzed modes.

Iterating the recursion yields

$$\boldsymbol{u}_{t'} = \mathbf{A}_t^k\boldsymbol{u}_t + \sum_{j=0}^{k-1}\mathbf{A}_t^j\mathbf{G}\nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t) + \sum_{j=0}^{k-1}\mathbf{A}_t^j\mathbf{G}\boldsymbol{\xi}_{t'-1-j}.$$

Taking expectation conditional on $\mathcal{S}$ eliminates the noise terms and gives the mean recursion in Equation 15.

For the covariance recursion, let $\mathbf{\Omega}_s = \mathrm{Cov}(\boldsymbol{u}_s \mid \mathcal{S})$. Under the martingale-difference property of the mini-batch noise (as used in the proof of Lemma 4.4), the cross-covariance terms vanish when propagating second moments across time, yielding

$$\mathbf{\Omega}_{t'} = \mathbf{A}_t^k\mathbf{\Omega}_t(\mathbf{A}_t^k)^\top + \sum_{j=0}^{k-1}\mathbf{A}_t^j\mathbf{G}\mathbf{C}_t\mathbf{G}^\top(\mathbf{A}_t^j)^\top,$$

which is exactly Equation 16. □

Corollary: Modal Recursions under Commutativity. To connect the matrix recursion in Lemma 4.5 to the scalar (per-direction) quantities used in Theorem 4.6 and in our empirical estimation procedure, we project the $2m$-dimensional augmented dynamics onto the eigenspace of the local Hessian. Following the stationary analysis, we adopt the same (approximate) commutativity assumption within the window: $\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)$ and the finite covariance $\mathbf{C}_t$ are simultaneously diagonalizable, so each stable eigendirection evolves independently.

Modal covariance recursion (commuting approximation). Let $(\lambda_i, \boldsymbol{e}_i)$ be an eigenpair of $\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)$ and define the (modal) noise variance $\gamma_i := \boldsymbol{e}_i^\top \mathbf{C}_t \boldsymbol{e}_i$. The corresponding 2×2 projected transition matrix is

$$\mathbf{A}_i = \begin{bmatrix} 1 - \eta\lambda_i & -\eta\kappa \\ \lambda_i & \kappa \end{bmatrix}, \tag{41}$$

with projected noise injection $\mathbf{Q}_i = \gamma_i \, \mathbf{G}_i \mathbf{G}_i^\top$ where $\mathbf{G}_i = [-\eta \quad 1]^\top$. Let $\mathbf{\Omega}_{t,i} \in \mathbb{R}^{2\times 2}$ denote the modal augmented covariance and define $\sigma_{t,i}^2 := [\mathbf{\Omega}_{t,i}]_{11}$. Iterating the projected recursion yields

$$\mathbf{\Omega}_{t+k,i} = \mathbf{A}_i^k \mathbf{\Omega}_{t,i} (\mathbf{A}_i^\top)^k + \sum_{j=0}^{k-1} \mathbf{A}_i^j \mathbf{Q}_i (\mathbf{A}_i^\top)^j, \quad \sigma_{t+k,i}^2 = [\mathbf{\Omega}_{t+k,i}]_{11}. \tag{42}$$

Stability requires $0 < \eta\lambda_i < 2(1+\kappa)$ for all analyzed modes i and for each learning rate used inside the window; when satisfied, the propagated term contracts at rate $O(\rho(\mathbf{A}_i)^{2k})$. If the window begins immediately after a learning-rate decay $\eta_1 \to \eta$, then $\sigma_{t,i}^2$ can be initialized by the stationary formula under η_1 using Equation 14 and evolved forward under the new learning rate η using Equation 42.

Non-stationary mean recursion (projected drift). Under the within-window invariance assumption in Lemma 4.5, the same affine dynamics yield an explicit short-window recursion for the mean drift. Define the *conditional* mean drift within the window $\Delta\bar{\boldsymbol{w}}_{t+k|t} := \mathbb{E}[\boldsymbol{w}_{t+k} \mid \mathcal{F}_t, \mathcal{S}] - \tilde{\boldsymbol{w}}_t$. Then

$$\Delta\bar{\boldsymbol{w}}_{t+k|t} = \mathbf{\Pi} \mathbf{A}_t^k \boldsymbol{u}_t + \sum_{j=0}^{k-1} \mathbf{\Pi} \mathbf{A}_t^j \mathbf{G} \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t), \tag{43}$$

where $\mathbf{\Pi} = [\mathbf{I}\ \mathbf{0}]$ is the projection onto the parameter subspace and $\boldsymbol{u}_t = [\boldsymbol{w}_t - \tilde{\boldsymbol{w}}_t; \boldsymbol{v}_t]$ is the augmented state. Taking expectation conditional on $\mathcal{S}$ yields the *posterior* mean drift $\Delta\bar{\boldsymbol{w}}_{t+k} := \boldsymbol{\mu}_{t+k} - \tilde{\boldsymbol{w}}_t$ with $\boldsymbol{\mu}_{t+k} = \mathbb{E}[\boldsymbol{w}_{t+k} \mid \mathcal{S}]$:

$$\Delta\bar{\boldsymbol{w}}_{t+k} = \mathbf{\Pi} \mathbf{A}_t^k \bar{\boldsymbol{u}}_t + \sum_{j=0}^{k-1} \mathbf{\Pi} \mathbf{A}_t^j \mathbf{G} \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t), \tag{44}$$

where $\bar{\boldsymbol{u}}_t = \mathbb{E}[\boldsymbol{u}_t \mid \mathcal{S}] = [\boldsymbol{\mu}_t - \tilde{\boldsymbol{w}}_t; \bar{\boldsymbol{v}}_t]$ with $\boldsymbol{\mu}_t = \mathbb{E}[\boldsymbol{w}_t \mid \mathcal{S}]$ and $\bar{\boldsymbol{v}}_t = \mathbb{E}[\boldsymbol{v}_t \mid \mathcal{S}]$.

B.3 CLOSED-FORM BOUND: PROOF OF THEOREM 4.6

Proof of Theorem 4.6. We divide the proof of Theorem 4.6 into three steps.

Step 1: Start from the compact PAC-Bayesian bound. Fix a timestep t and consider the posterior $\mathcal{Q}_t = \mathcal{N}(\boldsymbol{\mu}_t, \mathbf{\Sigma}_t)$ induced by momentum SGD. Applying Theorem 4.3 with reference point $\tilde{\boldsymbol{w}}_t$ gives

$$\begin{aligned} \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}_t}\big[\mathcal{L}_\epsilon(\boldsymbol{w})\big] \le{}& \hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t) + \big\langle \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t), \boldsymbol{\mu}_t - \tilde{\boldsymbol{w}}_t \big\rangle + \frac{1}{2}(\boldsymbol{\mu}_t - \tilde{\boldsymbol{w}}_t)^\top \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)(\boldsymbol{\mu}_t - \tilde{\boldsymbol{w}}_t) \\ &+ \frac{1}{2}\mathrm{Tr}\big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)\mathbf{\Sigma}_t\big) + \frac{1}{2\beta\sigma_{\mathcal{P}}^2}\Big(\|\boldsymbol{\mu}_t\|_2^2 + \mathrm{Tr}(\mathbf{\Sigma}_t)\Big) - \frac{1}{2\beta}\ln\det(\mathbf{\Sigma}_t) + C_{\mathrm{PB}} + R_t. \end{aligned} \tag{45}$$

Step 2: Matrix form and spectral specialization. Equation 17 follows directly in matrix form from Equation 45 after collecting the posterior-independent constants into c_{PB}. Under the additional simultaneous-diagonalization condition $\mathbf{H}_t\mathbf{\Sigma}_t = \mathbf{\Sigma}_t\mathbf{H}_t$, the trace terms admit the spectral representation

$$\mathrm{Tr}\big(\hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)\mathbf{\Sigma}_t\big) = \sum_{i=1}^m \lambda_{t,i}\,\sigma_{t,i}^2, \quad \ln\det(\mathbf{\Sigma}_t) = \sum_{i=1}^m \ln\sigma_{t,i}^2, \quad \mathrm{Tr}(\mathbf{\Sigma}_t) = \sum_{i=1}^m \sigma_{t,i}^2.$$

The covariance-trace KL contribution then combines with the curvature-weighted variance as:

$$\frac{1}{2}\mathrm{Tr}(\mathbf{H}_t\mathbf{\Sigma}_t) + \frac{1}{2\beta\sigma_{\mathcal{P}}^2}\mathrm{Tr}(\mathbf{\Sigma}_t) = \frac{1}{2}\sum_{i=1}^m \left(\lambda_{t,i} + \frac{1}{\beta\sigma_{\mathcal{P}}^2}\right)\sigma_{t,i}^2.$$

This is the spectral specialization used for dominant-mode diagnostics, not an extra requirement for the matrix-form bound.

Step 3: Stationary vs. non-stationary posterior parameters. Finally, the bound in Equation 17 is made explicit by plugging in the posterior parameters. In the stationary regime, Lemma 4.4 provides the closed-form mean and covariance (Equations 13 and 14). In particular, letting $\boldsymbol{g}_t := \nabla\hat{\mathcal{L}}_\epsilon(\tilde{\boldsymbol{w}}_t)$ and $\mathbf{H}_t := \hat{\mathbf{H}}_\epsilon(\tilde{\boldsymbol{w}}_t)$, the stationary mean drift $\Delta\bar{\boldsymbol{w}}_t = \boldsymbol{\mu}_t - \tilde{\boldsymbol{w}}_t = -\mathbf{H}_t^{-1}\boldsymbol{g}_t$ implies the familiar Newton-type optimization gain

$$\langle \boldsymbol{g}_t, \Delta\bar{\boldsymbol{w}}_t \rangle \;+\; \frac{1}{2}\Delta\bar{\boldsymbol{w}}_t^\top \mathbf{H}_t\, \Delta\bar{\boldsymbol{w}}_t \;=\; -\frac{1}{2}\boldsymbol{g}_t^\top \mathbf{H}_t^{-1}\boldsymbol{g}_t.$$

In the non-stationary regime, Lemma 4.5 provides the windowed matrix recursion for $(\boldsymbol{\mu}_{t+k}, \boldsymbol{\Omega}_{t+k})$, and the modal recursion in Appendix B.2 (Equations 41 and 42) yields the corresponding scalar variances $\{\sigma^2_{t+k,i}\}_{i=1}^m$ under commutativity. Substituting these quantities into Equation 17 completes the proof. □

C PROOFS OF THEORETICAL RESULTS IN SECTION 3 & SECTION 5

C.1 PROOF OF THEOREM 3.2

Theorem 3.2 is stated for bounded losses because Catoni's bounded-difference concentration is used in the generic PAC-Bayesian step. This condition is satisfied directly by bounded surrogate losses and clipped cross-entropy. For the cross-entropy losses used in our experiments, one can replace the bounded-loss concentration step by PAC-Bayes-Chernoff bounds for unbounded losses under light-tail or moment assumptions [Casado et al., 2024]. This changes concentration constants or tail terms but not the Section 4 posterior-averaged loss and Gaussian-KL decomposition.

To prove Theorem 3.2, we first state a general PAC-Bayes inequality, also known as Catoni's bound [Catoni, 2007].

Lemma C.1 (PAC-Bayes Bound (Catoni's bound))**.** *Let $\alpha \in (0,1)$, $\beta > 0$, and $\mathcal{D}$ be any distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{W}$ be a parameter space and $\mathcal{P}$ be a data-independent prior on $\mathcal{W}$. Consider any measurable loss $\ell : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to [0, C_\ell]$ bounded by the scalar $C_\ell > 0$. Given an i.i.d. sample set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{S}|}$ drawn from $\mathcal{D}$, for any posterior $\mathcal{Q}$ on $\mathcal{W}$, with probability at least $1-\alpha$ over the draw of $\mathcal{S}$, we have*

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\ell(\boldsymbol{w};\boldsymbol{x},y)\big] \;\le\; \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[\frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}}\ell(\boldsymbol{w};\boldsymbol{x},y)\right] + \frac{\beta C_\ell^2}{8|\mathcal{S}|} + \frac{1}{\beta}\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) - \frac{1}{\beta}\ln\alpha.$$

Proof of Theorem 3.2. We instantiate Lemma C.1 with the adversarial loss

$$\tilde{\ell}(\boldsymbol{w};\boldsymbol{x},y) \;:=\; \ell_{\mathrm{adv}}(\boldsymbol{w},\boldsymbol{x},y) = \max_{\boldsymbol{\delta}\in B_\varepsilon(0)} \ell(\boldsymbol{w};\, \boldsymbol{x}+\boldsymbol{\delta}, y)\,,$$

where the base loss ℓ is bounded by C_ℓ and the perturbation set $B_\varepsilon(0)$ is fixed. Since $\ell \in [0, C_\ell]$, the maximization preserves boundedness, so $\tilde{\ell} \in [0, C_\ell]$. Measurability follows directly from that of ℓ and the continuity of $(\boldsymbol{x},\boldsymbol{\delta}) \mapsto \ell(\boldsymbol{w};\boldsymbol{x}+\boldsymbol{\delta},y)$. Applying Lemma C.1 to $\tilde{\ell}$ yields

$$\mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\ell_{\mathrm{adv}}(\boldsymbol{w},\boldsymbol{x},y)\big] \;\le\; \mathbb{E}_{\boldsymbol{w}\sim\mathcal{Q}}\left[\frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}}\ell_{\mathrm{adv}}(\boldsymbol{w},\boldsymbol{x},y)\right] + \frac{\beta C_\ell^2}{8|\mathcal{S}|} + \frac{1}{\beta}\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) - \frac{1}{\beta}\ln\alpha.$$

Plugging the definitions of empirical and expected adversarial losses and rearranging the terms completes the proof. □

C.2 PROOF OF PROPOSITION 5.1

Proof of Proposition 5.1. We prove the proposition in terms of the two commonly adopted perturbation metrics in the prior literature on adversarial training: L_2 norm and L_∞ norm. In particular, our derivations rely on the assumption that the adversarial loss can be locally linearized for each training example (e.g., when the perturbation strength ϵ is small enough).

L_2 Case. We start with the simpler case of L_2 perturbations. Consider the adversarial training objective $\hat{\mathcal{L}}_\epsilon$ in Equation 2 with $p = 2$. Assuming the adversarial loss can be locally linearized at each $(\boldsymbol{x}, y)$, we can show that

$$\hat{\mathcal{L}}_\epsilon(\boldsymbol{w}; L_2) = \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}}\left[\ell(\boldsymbol{w},\boldsymbol{x},y) + \max_{\|\boldsymbol{\delta}\|_2\le\epsilon}\langle\boldsymbol{\delta}, \nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\rangle\right] = \hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}}\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2, \tag{46}$$

where $\hat{\mathcal{L}}(\boldsymbol{w}) = \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \ell(\boldsymbol{w},\boldsymbol{x},y)$ denotes the clean loss, and the second equality holds due to Cauchy–Schwarz inequality. Equation 46 suggests that achieving low empirical adversarial loss requires: (i) a good fit of the clean examples (i.e., low $\hat{\mathcal{L}}(\boldsymbol{w})$), and (ii) small input gradient sensitivity across the training dataset (i.e., low $\|\nabla_x \ell(\boldsymbol{w},\boldsymbol{x},y)\|_2$).

Taking the derivative of Equation 46 with respect to the model weights gives the gradient of the adversarial loss:

$$\begin{aligned}\nabla_{\boldsymbol{w}}\hat{\mathcal{L}}_\epsilon(\boldsymbol{w};L_2) &= \nabla_{\boldsymbol{w}}\hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \nabla_{\boldsymbol{w}}\Big(\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2\Big) \\ &= \nabla_{\boldsymbol{w}}\hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2^{-1}\Big(\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\Big)^\top \nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y), \end{aligned} \tag{47}$$

where the second equality follows from the chain rule. Taking another derivative with respect to the weight parameters of Equation 47 gives us the connection between the Hessian in the parameter space and the input gradient sensitivity:

$$\begin{aligned}\nabla^2_{\boldsymbol{w}}\hat{\mathcal{L}}_\epsilon(\boldsymbol{w};L_2) &= \nabla^2_{\boldsymbol{w}}\hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \nabla_{\boldsymbol{w}}\Big(\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2^{-1}\Big(\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\Big)^\top \nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\Big) \\ &= \underbrace{\nabla^2_{\boldsymbol{w}}\hat{\mathcal{L}}(\boldsymbol{w})}_{\text{clean Hessian}} + \frac{\epsilon}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \underbrace{\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2^{-1}\cdot\Big(\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\Big)^\top \nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)}_{\text{PSD cross-Hessian term}} + \mathcal{R}(\boldsymbol{w}), \end{aligned} \tag{48}$$

where $\mathcal{R}(\boldsymbol{w})$ stands for the remaining negligible terms, involving a third-order derivative term and negative rank-1 correction from norm normalization. One can expect the remaining terms to have a limited effect on the large Hessian eigenvalues compared to the two terms explicitly specified in Equation 48. Compared with the clean counterpart, the Hessian of the adversarial training loss includes a positive semi-definite (PSD) term averaged across the training dataset, involving the cross Hessian $\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)$, which captures how the input gradient of adversarial loss changes with $\boldsymbol{w}$.

Note that given any direction in parameter space $\boldsymbol{v}\in\mathbb{R}^m$, we have

$$\boldsymbol{v}^\top\Big(\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2^{-1}\cdot\Big(\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\Big)^\top \nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\Big)\boldsymbol{v} = \frac{\big\|\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\boldsymbol{v}\big\|_2^2}{\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_2} \geq 0. \tag{49}$$

Equation 49 suggests that: (i) the curvature of adversarial loss is always expected to be larger than that of clean loss due to the extra PSD term, which aligns with our empirical observations that the top Hessian eigenvalues of adversarially trained models are larger than those of standard training, and (ii) if the top eigenvalues of the cross Hessian grow or the ℓ_2 norm of the input gradient reduces, the curvature of the adversarial loss will accordingly grow.

L_∞ Case. Now, we turn to prove Proposition 5.1 under L_∞-norm bounded perturbations. Consider Equation 2 with $p=\infty$. Under the same local loss linearization assumption, according to Hölder's inequality, we have

$$\hat{\mathcal{L}}_\epsilon(\boldsymbol{w};L_\infty) = \hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|}\sum_{(x,y)\in\mathcal{S}} \big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big\|_1. \tag{50}$$

Similar to the L_2 case, we next compute the first-order derivative of the adversarial loss with respect to $\boldsymbol{w}$:

$$\nabla_{\boldsymbol{w}}\hat{\mathcal{L}}_\epsilon(\boldsymbol{w};L_\infty) = \nabla_{\boldsymbol{w}}\hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \big(\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big)^\top \operatorname{sgn}(\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)),$$

where $\operatorname{sgn}(\cdot)$ denotes the sign operator. For simplicity, denote by $g(\boldsymbol{w},\boldsymbol{z}) = \nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)$ the input gradient at $\boldsymbol{z}=(\boldsymbol{x},y)$ in the following derivation. A key difference from the L_2 case is how to compute the second-order derivative given the sign operator. Instead of directly differentiating through the sign operator, one can make use of the following fact:

$$\boldsymbol{\delta}^* = \epsilon\cdot\operatorname{sgn}\big(\nabla_{\boldsymbol{x}}\ell(\boldsymbol{w},\boldsymbol{x},y)\big) = \underset{\|\boldsymbol{\delta}\|_\infty\leq\epsilon}{\arg\max}\ \big\langle g(\boldsymbol{w},\boldsymbol{z}),\boldsymbol{\delta}\big\rangle,$$

Note that it is a linear programming (LP) in terms of $\boldsymbol{\delta}$ with coefficient $g(\boldsymbol{w},\boldsymbol{z})$ and a fixed L_∞-norm ball. Taking the derivative with respect to $\boldsymbol{w}$ and using the chain rule of generalized derivatives, we obtain:

$$\frac{\partial\boldsymbol{\delta}^*}{\partial\boldsymbol{w}} = \mathbf{D}(\boldsymbol{w},\boldsymbol{z})\cdot\nabla_{\boldsymbol{w}}g(\boldsymbol{w},\boldsymbol{z}),$$

where $\mathbf{D}(\boldsymbol{w}, \boldsymbol{z}) = \mathrm{diag}(d_1, d_2, \ldots, d_m)$ is a diagonal matrix that encodes the active set of coordinates: $d_i = 1$ if the i-th coordinate $[g(\boldsymbol{w}, \boldsymbol{z})]_i = 0$ (sign-switching region, constraint is inactive), and $d_i = 0$ if $[g(\boldsymbol{w}, \boldsymbol{z})]_i \neq 0$ (locally stable sign, saturated coordinate). The above equation results from the classical LP sensitivity theory [Rockafellar and Wets, 1998], where $\mathbf{D}$ is essentially an active-set selection matrix arising from the generalized derivative of the LP solution map.

Now, we can derive the second-order derivative with respect to the adversarial loss under L_∞ perturbations:

$$\begin{aligned}\nabla_w^2 \hat{\mathcal{L}}_\epsilon(\boldsymbol{w}; L_\infty) &= \nabla_{\boldsymbol{w}}^2 \hat{\mathcal{L}}(\boldsymbol{w}) + \frac{\epsilon}{|\mathcal{S}|} \sum_{\boldsymbol{z} \in \mathcal{S}} \big(\nabla_{\boldsymbol{w}} g(\boldsymbol{w}, \boldsymbol{z})\big)^\top \mathbf{D}(\boldsymbol{w}, \boldsymbol{z}) \big(\nabla_{\boldsymbol{w}} g(\boldsymbol{w}, \boldsymbol{z})\big) + \mathcal{R}(\boldsymbol{w}) \\ &= \underbrace{\nabla_{\boldsymbol{w}}^2 \hat{\mathcal{L}}(\boldsymbol{w})}_{\text{clean Hessian}} + \frac{\epsilon}{|\mathcal{S}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \underbrace{\Big(\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y)\Big)^\top \mathbf{D}(\boldsymbol{w}, \boldsymbol{x}, y) \Big(\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y)\Big)}_{\text{PSD cross-Hessian term}} + \mathcal{R}(\boldsymbol{w}), \end{aligned} \tag{51}$$

where $\mathcal{R}(w)$ stands for the remainder terms, such as the higher-order derivatives. Comparing Equation 51 to Equation 49, we note that the difference for L_∞ is that the PSD cross-Hessian term is controlled by the diagonal active-set projector $\mathbf{D}(\boldsymbol{w}, \boldsymbol{x}, y)$, whereas the cross-Hessian term for L_2 case is scaled by a smooth projector $1/\|\nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y)\|_2$. Finally, for any direction $\boldsymbol{v} \in \mathbb{R}^m$ in the parameter space, we have

$$\boldsymbol{v}^\top \Big(\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y)\Big)^\top \mathbf{D}(\boldsymbol{w}, \boldsymbol{x}, y) \Big(\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y)\Big) \boldsymbol{v} = \Big\| \mathbf{D}(\boldsymbol{w}, \boldsymbol{x}, y)^{1/2} \cdot \nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y) \boldsymbol{v} \Big\|_2^2 \geq 0,$$

which immediately suggests that the cross-Hessian term is positive semi-definite (PSD).

Why adversarial training increases loss curvature. Now we can explain why the adversarial training objective encourages an increase in the top Hessian eigenvalues. First, optimizing the empirical adversarial loss reduces the average input gradient sensitivity $\frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \|\nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y)\|_2$ by design (specifically for the L_2 case); otherwise, the second term in Equation 46 will not be well-controlled. Second, note that robust features are known to be highly anisotropic, whereas standard features are often noisier; effective adversarial training requires aligning input gradients across training examples within the robust feature subspace, which also suggests the same for the parameter gradients along the optimization trajectory. This implies that input and parameter gradients will become increasingly aligned as adversarial training advances.

Note that the cross-Hessian term for each individual data point can be expressed as:

$$\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{w}, \boldsymbol{x}, y) = \frac{\partial^2 \ell}{\partial f^2} (\nabla_{\boldsymbol{w}} f)(\nabla_{\boldsymbol{x}} f)^\top + \frac{\partial \ell}{\partial f} \nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{x}} f, \;\; \text{where } \ell(\boldsymbol{w}, \boldsymbol{x}, y) = \ell(f_{\boldsymbol{w}}(\boldsymbol{x}), y). \tag{52}$$

Near a well-fitted solution, we know $\frac{\partial \ell}{\partial f} \to 0$. This suggests that the second term in Equation 52 is negligible compared to the first term, which is essentially the outer product between input and parameter gradient (rank 1 per sample). Enhanced alignment reinforces the outer-product growth and eventually boosts the spectral concentration of the cross Hessian. According to Equations 49 and 51, increasing the perturbation size ϵ will naturally boost the Hessian eigenvalues, since the second term is proportional to ϵ. A larger ϵ suggests a stricter alignment requirement for both input and parameter gradients, which can further enhance the spectral concentration with respect to the cross-Hessian term. The above interpretations support our observations in Figure 1: (i) ST can attain very low clean training loss and small Hessian eigenvalues simultaneously; (ii) AT lowers robust training loss in later stages when η is small, aligning gradients within the robust feature subspace, which inevitably boosts the top Hessian eigenvalues; and (iii) AWP optimizes for worst-case weight perturbations and effectively controls curvature growth, but the sharpness penalization can weaken gradient alignment in the robust feature subspace (e.g., the 20-th eigenvalue λ_{20} remains constant), resulting in a smaller decrease in robust training loss. □

D DETAILED EXPERIMENTAL SETUP

D.1 DATASETS AND TRAINING BASELINES

We conduct experiments on CIFAR-10 [Krizhevsky et al., 2009] using the standard train/test split. Inputs are scaled to $[0, 1]$ and normalized channel-wise. Data augmentation includes random cropping with 4-pixel padding and random horizontal flipping. We also evaluate on CIFAR-100 [Krizhevsky et al., 2009] and SVHN [Netzer et al., 2011] to verify the generality of our findings. Unless stated otherwise, we use PreActResNet-18 [He et al., 2016] and optimize with momentum SGD ($\kappa = 0.9$) for 200 epochs with batch size 128 and weight decay 5×10^{-4}. For CIFAR-10 (and CIFAR-100), we use an

initial learning rate of 0.1 and a piecewise schedule decayed by a factor of 10 at epochs 100 and 150. For evaluating model robustness, we adopt PGD adversarial training (AT) [Madry et al., 2018] as the default baseline. We generate ℓ_∞ adversarial perturbations with $\epsilon = 8/255$, step size $2/255$, 10 attack iterations, and 1 random restart.

For AWP [Wu et al., 2020], we follow the standard configuration with perturbation radius $\gamma_{\mathrm{AWP}} = 0.01$ (no warmup) and use the same outer-loop optimizer and learning-rate schedule. Unless stated otherwise, robust metrics are computed via PGD under the same ϵ; we use 10 steps for AT and 20 steps for AWP. All experiments were run on NVIDIA A100 GPUs, using a single GPU per training/estimation job. Based on the wall-clock times recorded in our training logs and the end-to-end per-epoch estimation pipeline, reproducing all results and figures in this paper requires approximately ~ 220 GPU-hours in total (about ~ 65 GPU-hours for training and ~ 155 GPU-hours for spectral estimation).

D.2 SPECTRAL ESTIMATION PROTOCOL

For computational efficiency, we approximate curvature and gradient-noise statistics using a small number of randomly sampled mini-batches. Specifically, unless stated otherwise, we estimate the top-k Hessian eigenpairs with $k = 20$ by averaging the loss over $M = 128$ mini-batches, performing 30 power iterations for each eigenpair, and computing the projected gradient-noise covariances from the same $M = 128$ mini-batches. This stochastic approximation is sufficient to capture the dominant spectral structure while keeping runtime feasible. Algorithm 1 summarizes the empirical procedure used to estimate the spectral statistics and bound components reported in Section 5. To obtain the Hessian spectrum, we approximate Hessian–vector products (HVPs) via the identity

$$\hat{\mathbf{H}}_\epsilon(\boldsymbol{w})\boldsymbol{v} \;=\; \nabla_{\boldsymbol{w}}\Big(\nabla_{\boldsymbol{w}}\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})^\top \boldsymbol{v}\Big),$$

which can be computed efficiently by automatic differentiation without explicitly forming $\hat{\mathbf{H}}_\epsilon$. We then apply power iteration with Gram–Schmidt orthogonalization to extract the top-k eigenvectors $\{\boldsymbol{v}_i\}$, and estimate their associated eigenvalues using the Rayleigh quotient,

$$\lambda_i \;=\; \frac{\boldsymbol{v}_i^\top(\hat{\mathbf{H}}_\epsilon(\boldsymbol{w})\boldsymbol{v}_i)}{\boldsymbol{v}_i^\top \boldsymbol{v}_i}.$$

This procedure is consistent with established practice for large-scale curvature estimation [Dangel et al., 2020, Yao et al., 2020]. For the posterior structure, we consider the stochastic gradients at the mini-batch level, $\boldsymbol{g}_b = \nabla_{\boldsymbol{w}}\mathcal{L}_b(\boldsymbol{w})$. Projecting these gradients onto the subspace spanned by the leading Hessian eigenvectors $\mathbf{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$ yields the projected quantities $\boldsymbol{p}_b = \mathbf{V}^\top \boldsymbol{g}_b \in \mathbb{R}^k$. Their covariance matrix is $\Gamma = \mathrm{Cov}[\boldsymbol{p}_b]$, whose diagonal entries $\gamma_i = \Gamma_{ii}$ quantify the variance of stochastic gradients along the principal curvature directions.

By construction, this definition ensures that λ_i characterizes curvature while γ_i represents the corresponding noise magnitude in the same eigendirections. This approach follows recent empirical observations that gradient-noise covariance tends to align with the Hessian eigenspectrum in neural networks [Jastrzebski et al., 2019, Ziyin et al., 2025], allowing for a direct analysis of curvature–noise interactions. When instantiating our closed-form formulas (e.g., Equation 14), we further adopt a diagonal approximation in the Hessian eigenspace, using only the diagonal entries $\{\gamma_i\}$ of Γ. This approximation is exact under the commutativity assumption (simultaneous diagonalizability of $\mathbf{H}$ and $\mathbf{C}$), and its accuracy in practice depends on the magnitude of the off-diagonal energy in Γ, which quantifies curvature–noise misalignment within the top-k subspace. We empirically verify that both commutativity and alignment hold to a good approximation throughout training under both AT and AWP (Appendix F.1), justifying the use of $\{\lambda_i, \gamma_i\}$ as matched curvature–noise pairs in our empirical decomposition.

Negative eigenvalues. The stationary covariance formula (Equation 14) requires local convexity ($\mathbf{H} \succ 0$); negative eigenvalues indicate saddle-point geometry where the stationary Gaussian approximation breaks down. In practice, when a few leading directions have negative curvature, we exclude them from the stationary instantiation and attribute their effect to the non-stationary drift analysis.

D.3 TRACKING STATIONARY AND TRANSIENT REGIMES

When logging full-space stationary bias proxies, we estimate the damped Newton direction by conjugate gradient, solving $(\hat{\mathbf{H}}_\epsilon(\boldsymbol{w}) + \lambda_{\mathrm{damp}}\mathbf{I})\boldsymbol{m} = -\nabla\hat{\mathcal{L}}_\epsilon(\boldsymbol{w})$ with $\lambda_{\mathrm{damp}} = 10^{-3}$, a maximum of 200 CG iterations, and stopping tolerance 10^{-6}. To instantiate the non-stationary transient dynamics, we treat each learning-rate change (epochs where the logged step size η differs from the previous epoch) as the onset of a transient. Starting from the stationary solution at the pre-change epoch, we

iterate the per-mode 2×2 recursions for the number of SGD steps per epoch (computed from dataset size and batch size) and apply the resulting non-stationary overrides epoch-wise until they converge back to the stationary formulas. Convergence is detected via the symmetric relative error between stationary and iterated per-mode quantities, using a threshold $\tau = 0.05$ for $P = 3$ consecutive epochs, after which we revert to the stationary estimates until the next learning-rate change. To account for cross-epoch drift of the empirically estimated top-k eigenspace, we align consecutive eigenbases by orthogonal Procrustes and transport diagonal mode quantities by the mixing map $x \leftarrow \mathrm{diag}(\mathbf{R}\mathrm{diag}(x)\mathbf{R}^\top) = (\mathbf{R} \odot \mathbf{R})x$ before applying each epoch-level transient update.

D.4 RUNTIME AND SCALABILITY

The spectral-estimation protocol is an offline checkpoint diagnostic that avoids materializing the full Hessian and the full noise covariance. For T_{eval} evaluated checkpoints, k retained modes, n_{PI} power iterations, and M sampled mini-batches, the leading HVP cost is $O(T_{\mathrm{eval}} \cdot kM \cdot n_{\mathrm{PI}})$ backward-equivalent operations, plus $O(T_{\mathrm{eval}} \cdot M)$ mini-batch gradient evaluations for projected noise covariance. The WRN-34-10 study (Figure 9 in Appendix F.3) uses the same protocol, but with a smaller retained eigenspace. Scaling to ViT-scale models would require memory-efficient HVPs, activation checkpointing, or sketched leading-mode estimators, but the diagnostic remains checkpoint-based rather than a training-time intervention.

E THEORETICAL JUSTIFICATION FOR TOP-k PROXY

This section provides a theoretical justification for the top-k estimation used in our main experiments. In Section 5, note that we report top-k proxies for the spectral terms (e.g., $-\sum_i \ln \sigma_i^2$) that appear in our PAC-Bayes decomposition. These proxies are not intended to approximate the full m-dimensional quantities in absolute scale; rather, we use them to track the *temporal trends* during training. The following proposition formalizes this "dominant-mode proxy" viewpoint: the epoch-to-epoch *change* of the full entropy term is well-approximated by a top-k truncation whenever the tail modes are approximately stationary (as in a spiked-spectrum regime).

Proposition E.1 (Dominant-mode control of temporal trends)**.** *Let $\{\sigma_i^2(t)\}_{i=1}^m$ denote the eigenvalues of the posterior covariance at epoch t, and define the full and truncated entropy contributions*

$$E(t) := -\sum_{i=1}^{m} \ln \sigma_i^2(t), \qquad E_k(t) := -\sum_{i=1}^{k} \ln \sigma_i^2(t).$$

Then for any two epochs t, t',

$$\Big|(E(t') - E(t)) - (E_k(t') - E_k(t))\Big| \;\le\; \sum_{i=k+1}^{m} \left|\ln \frac{\sigma_i^2(t')}{\sigma_i^2(t)}\right|. \tag{53}$$

Moreover, if the tail modes satisfy a small relative-change condition $\left|\frac{\sigma_i^2(t')}{\sigma_i^2(t)} - 1\right| \le \varepsilon_i \le \frac{1}{2}$ for all $i > k$, then

$$\Big|(E(t') - E(t)) - (E_k(t') - E_k(t))\Big| \;\le\; 2 \sum_{i=k+1}^{m} \varepsilon_i. \tag{54}$$

Under the commuting stationary approximation in Equation 14, the per-mode variance satisfies

$$\sigma_i^2(t) = \frac{\eta(t)}{1-\kappa} \cdot \frac{\gamma_i(t)}{\lambda_i(t)\left(2 - \frac{\eta(t)}{1+\kappa}\lambda_i(t)\right)}. \tag{55}$$

Thus, the relative change in σ_i^2 (Equation 53) is controlled by relative changes in the matched spectral statistics (λ_i, γ_i).

Proof. We start from the decomposition

$$(E(t') - E(t)) - (E_k(t') - E_k(t)) = -\sum_{i=k+1}^{m} \big(\ln \sigma_i^2(t') - \ln \sigma_i^2(t)\big) = -\sum_{i=k+1}^{m} \ln \frac{\sigma_i^2(t')}{\sigma_i^2(t)}.$$

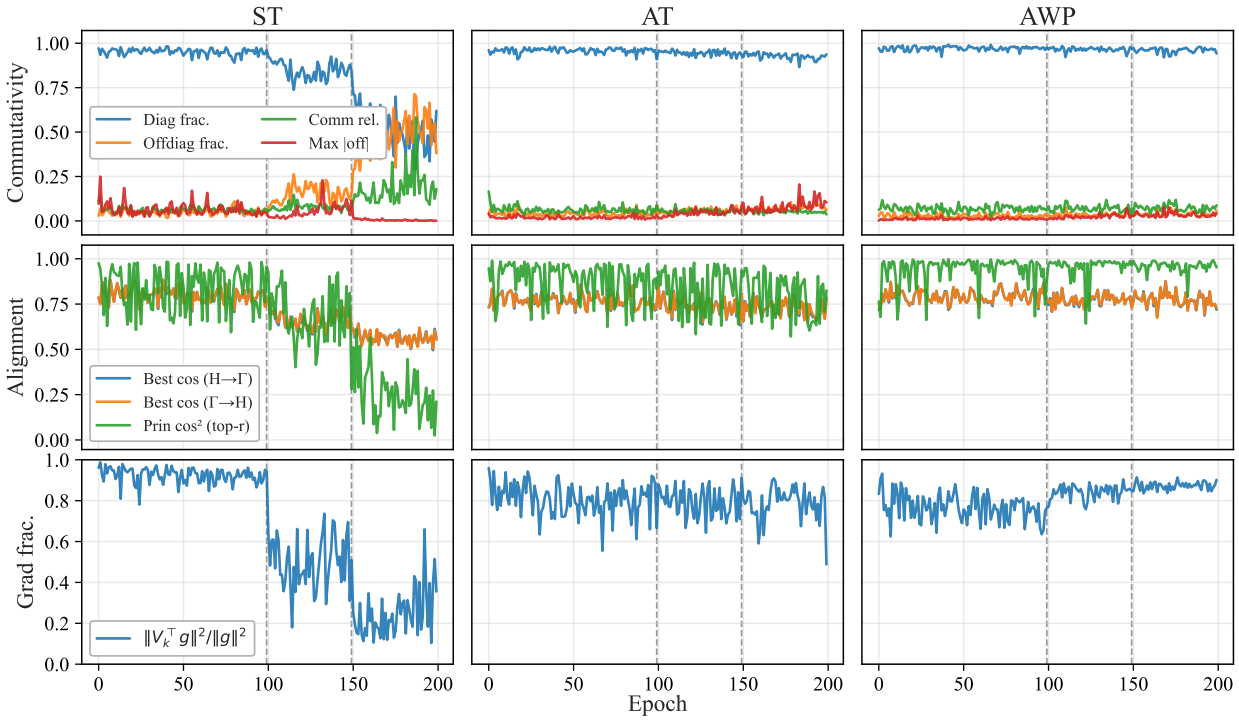

Figure 3: Commutativity, alignment, and top-k representativeness diagnostics (CIFAR-10; $k = 20$). Columns correspond to ST/AT/AWP, and vertical dashed lines indicate learning-rate drops. **Top row:** commutativity/diagonality of the projected covariance $\mathbf{\Gamma}_t$ in the Hessian basis. Legend abbreviations: *Diag frac.* $= \|\mathrm{diag}(\mathbf{\Gamma}_t)\|_F^2/\|\mathbf{\Gamma}_t\|_F^2$, *Offdiag frac.* $= 1-$Diag frac., *Comm rel.* $= \|[\mathbf{\Lambda}_t, \mathbf{\Gamma}_t]\|_F/(\|\mathbf{\Lambda}_t\|_F\|\mathbf{\Gamma}_t\|_F)$, and *Max |off|* $= \max_{i\neq j}|(\mathbf{\Gamma}_t)_{ij}|$, with $\mathbf{\Lambda}_t = \mathrm{diag}(\lambda_1(t), \ldots, \lambda_k(t))$. **Middle row:** alignment between the dominant eigenspaces of $\mathbf{\Gamma}_t$ and the Hessian eigenbasis (cosine similarities and a principal-angle proxy on the top-r subspace with $r = 3$). **Bottom row:** gradient-energy fraction in the Hessian top-k subspace, $\|\mathbf{V}_t^\top \boldsymbol{g}(t)\|^2/\|\boldsymbol{g}(t)\|^2$ (*Grad frac.*). Overall, AT and AWP maintain strong commutativity and alignment throughout training and retain high gradient-energy ratios (top-k remains representative for gradient-coupled quantities). ST is less stable, and its commutativity/alignment diagnostics can degrade in late training, alongside reduced gradient-energy ratios.

Applying the triangle inequality yields equation 53. For equation 54, write $\sigma_i^2(t')/\sigma_i^2(t) = 1 + r_i$ with $|r_i| \le 1/2$. Using the bound $|\ln(1+r)| \le 2|r|$ valid for $|r| \le 1/2$, we obtain $\left|\ln \frac{\sigma_i^2(t')}{\sigma_i^2(t)}\right| \le 2\varepsilon_i$ for any $i > k$, which completes the proof. Finally, Equation 55 is the eigendirection-wise specialization of Equation 14 under commutativity, showing explicitly that $\ln \sigma_i^2$ is an additive combination of $\ln \eta$, $\ln \gamma_i$, and functions of λ_i. □

F ADDITIONAL EXPERIMENTS

F.1 COMMUTATIVITY AND ALIGNMENT ASSUMPTIONS

Our theoretical instantiations rely on a *commuting/diagonal* approximation, which treats the projected gradient-noise covariance as approximately diagonal in the Hessian eigenbasis. We empirically probe this assumption—and the gradient-relevance of the Hessian top-k subspace—on CIFAR-10 for ST, AT, and AWP. The results are presented in Figure 3.

Measurement protocol. At each epoch t, we evaluate the empirical objective at the checkpoint $\boldsymbol{w}_t$ (clean loss for ST; robust loss for AT/AWP), estimate the top-k Hessian eigenpairs $\{(\lambda_i(t), \boldsymbol{v}_i(t))\}_{i=1}^k$, and form the matrix $\mathbf{V}_t = [\boldsymbol{v}_1(t), \ldots, \boldsymbol{v}_k(t)]$. Using mini-batch gradients $\boldsymbol{g}_b(t)$, we estimate the *projected* gradient-noise covariance $\mathbf{\Gamma}_t := \mathrm{Cov}_b(\mathbf{V}_t^\top \boldsymbol{g}_b(t)) \in \mathbb{R}^{k\times k}$, whose diagonal entries are the per-mode variances $\gamma_i(t)$ used in our diagonal instantiation. We then report: (a) commutativity/diagonality metrics based on the off-diagonal energy of $\mathbf{\Gamma}_t$ in the Hessian basis and the relative commutator norm $\|[\mathbf{\Lambda}_t, \mathbf{\Gamma}_t]\|_F/(\|\mathbf{\Lambda}_t\|_F\|\mathbf{\Gamma}_t\|_F)$ with $\mathbf{\Lambda}_t = \mathrm{diag}(\lambda_1(t), \ldots, \lambda_k(t))$; (b) alignment metrics based on cosine similarities and principal angles between the dominant eigenspaces of $\mathbf{\Gamma}_t$ and the Hessian basis; and (c) the fraction of gradient energy

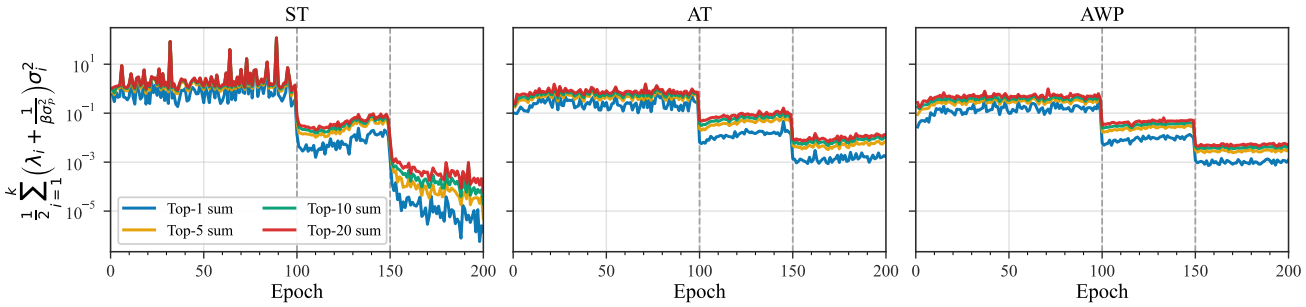

Figure 4: Full-KL shifted variance diagnostics for CIFAR-10 across ST, AT, and AWP (ℓ_∞, $\epsilon = 8/255$). Curves report top-k partial sums of $\frac{1}{2}\sum_{i=1}^{k}(\lambda_{t,i} + 1/(\beta\sigma_{\mathcal{P}}^2))\sigma_{t,i}^2$ for $k \in \{1, 5, 10, 20\}$; dashed lines mark learning-rate drops. This figure verifies that the covariance-trace KL shift included in Theorem 4.6 does not change the qualitative temporal diagnostics.

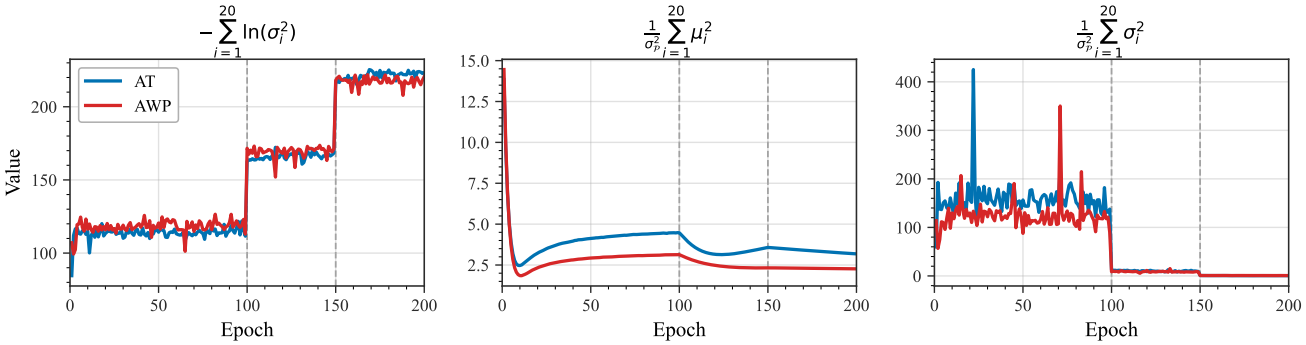

Figure 5: Posterior-dependent KL components for CIFAR-10 across AT and AWP (ℓ_∞, $\epsilon = 8/255$), using the top-20 retained modes. From left to right: log-determinant component, mean-norm component, and covariance-trace component.

captured by the Hessian top-k subspace, $\|\mathbf{V}_t^\top \boldsymbol{g}(t)\|^2/\|\boldsymbol{g}(t)\|^2$ (*Grad frac.*).

If the commutativity assumption breaks, then the two matrices cannot be simultaneously diagonalized, and the spectral quantities $\{\lambda_i, \gamma_i\}$ would no longer represent matched curvature–noise pairs. As a consequence, the posterior covariance could exhibit uncontrolled cross-terms, making the PAC-Bayesian bound less interpretable and potentially much looser. Similarly, if alignment were absent, the principal directions of stochastic gradient noise would not coincide with those of curvature. This mismatch would spread noise across directions with different curvatures, leading to inefficient exploration, less reliable stationary approximations, and weaker predictive power of our framework.

Top-k representativeness. In addition to commutativity/alignment, our use of top-k truncation implicitly assumes that the dominant subspace remains *gradient-relevant*. We therefore report the gradient-energy fraction captured by the Hessian top-k subspace, $\|\mathbf{V}_t^\top \boldsymbol{g}(t)\|^2/\|\boldsymbol{g}(t)\|^2$ (*Grad frac.*; bottom row of Figure 3). AT/AWP retain high ratios throughout training, whereas ST exhibits a more pronounced drop after learning-rate changes, indicating reduced top-k representativeness in late-stage ST. Taken together, these diagnostics support the use of the diagonal/top-k approximation as an informative model for AT and AWP, while also identifying regimes (notably late-stage ST) where commutativity/alignment and top-k representativeness degrade, and the approximation should be interpreted with caution.

F.2 FULL KL AND OMITTED-TERM DIAGNOSTICS

The strict bound in Theorem 4.6 contains KL terms beyond the log-determinant proxy emphasized in the main diagnostic grid. To ensure the rigor of our analysis and avoid overlooking bound components that may have a critical impact on adversarially robust generalization, we run additional experiments to analyze all KL-related terms and their relative importance. The results are reported in Figure 4 and Figure 5.

In particular, the covariance-trace part of the KL shifts the curvature-weighted variance:

$$\text{from} \quad \frac{1}{2}\sum_i \lambda_{t,i}\sigma_{t,i}^2 \quad \text{to} \quad \frac{1}{2}\sum_i \left(\lambda_{t,i} + \frac{1}{\beta\sigma_{\mathcal{P}}^2}\right)\sigma_{t,i}^2.$$

For CIFAR-10 with $|\mathcal{S}| = 50{,}000$, we use $\beta = \sqrt{|\mathcal{S}|}$ and the initialization-matched prior scale $\sigma_{\mathcal{P}}^2 \approx 4.93 \times 10^{-4}$, which gives $1/(\beta\sigma_{\mathcal{P}}^2) \approx 9.07$. Figure 4 visualizes the diagnostic curves with consideration of this extra KL term. The results suggest that the qualitative temporal trends are not affected by including the KL covariance-trace term, which can be absorbed by the more dominant curvature-weighted variance term in our analytical framework.

In Figure 5, we further visualize the diagnostic curves of the remaining KL-related terms (mean-norm component and covariance-trace component) for both AT and AWP. After the learning rate drops, both terms become much smaller than the log-determinant KL component, suggesting negligible influence on robust generalization/overfitting.

F.3 GENERALIZABILITY STUDY

To test the robustness of the diagnostic pattern, we repeat the unified diagnostic grids under variations of (i) dataset/learning algorithm, (ii) perturbation radius ϵ, (iii) batch size B, and (iv) architecture, objective, dataset, and learning-rate schedule. All plotted quantities and curve definitions match Figures 1 and 2 in the main text: *loss/error* (top row), leading Hessian eigenvalues $\{\lambda_i\}$ (second row), projected gradient-noise variances $\{\gamma_i\}$ (third row), and the resulting bound decomposition terms (variance/entropy/bias; bottom rows). Our goal is not only to document how these curves shift across settings, but also to connect the shifts back to the theory-motivated diagnostics: larger curvature is associated with smaller inferred posterior variances, stronger gradient noise is associated with larger γ_i, and robust-overfitting regimes coincide with learning-rate changes that upset this balance and produce posterior-collapse signatures.

Datasets and Learning Algorithms. Figure 6 extends our analysis to CIFAR-100, SVHN, and semi-supervised adversarial training (SSAT) [Carmon et al., 2019]. For CIFAR-100 and SVHN, we use the same epoch budget and learning-rate decay *strategy* as CIFAR-10: 200 epochs with the same piecewise decay points (epochs 100 and 150). We keep the optimizer (momentum SGD with $\kappa = 0.9$ and weight decay 5×10^{-4}) and batch size (128) fixed. We use an initial learning rate of 0.1 for CIFAR-100 and 0.01 for SVHN; other attack hyperparameters follow the CIFAR-10 setting unless stated otherwise (for SVHN, we use a smaller PGD step size $1/255$).

Relative to CIFAR-10, CIFAR-100 exhibits substantially larger leading curvature and projected noise in the dominant modes (larger λ_1 and γ_1), with correspondingly larger magnitudes in the PAC-Bayes proxy terms after learning-rate decays. The post-decay regime shows sharper transients and stronger late-stage degradation in robust test performance, consistent with more pronounced robust overfitting. In contrast, SVHN displays smaller λ and γ trajectories and a smoother post-decay evolution, and its bound components evolve more steadily with weaker evidence of abrupt contraction. SSAT typically lowers risk curves relative to the purely supervised AT baseline, but its geometry does not exhibit the same systematic curvature suppression as AWP; accordingly, its spectra and bound terms fall between standard AT and AWP-like behavior.

These cross-dataset differences are consistent with a margin-distribution view of robust optimization. Harder, fine-grained classification (CIFAR-100) keeps a larger fraction of training points near the adversarial decision boundary, so the inner maximization continues to surface high-sensitivity examples later into training. This sustains sharp Gauss–Newton curvature modes in the robust loss and increases the variability of projected gradients, raising both λ_i and γ_i in the dominant subspace. In our dynamical PAC-Bayes instantiation, larger curvature is associated with stronger posterior contraction and more pronounced posterior-collapse diagnostics, while larger noise inflates σ_i^2; after learning-rate decays reduce the step size, this balance can shift toward stronger posterior-contraction signatures.

SVHN is relatively easy for the same architecture: margins are larger, and the adversarial maximization is less likely to produce strongly boundary-adjacent updates, resulting in a smaller dominant spectrum and a more stable curvature–noise balance. Finally, SSAT introduces additional constraints (via unlabeled examples) that can improve generalization, but it does not explicitly penalize weight-space sharpness as AWP does; hence, while risk curves improve, sharp directions need not be systematically suppressed, and the bound decomposition remains closer to AT than to AWP.

Perturbation Radius. Perturbation radius ϵ indicates the strength of the inner maximization and thus affects the geometry of the robust objective. Figure 7 reports AT runs with $\epsilon \in \{2/255, 4/255, 12/255\}$ (the baseline $\epsilon = 8/255$ is shown in the AT column of Figures 1 and 2). As ϵ increases, the dominant curvature modes become systematically sharper (both λ_1 and the top-k averages increase), and the robust test loss/error worsens. Projected noise variances also increase, indicating that stronger attacks not only steepen the robust objective but also change the variability of mini-batch gradients in the sharp subspace. These spectral shifts propagate into the bound decomposition: larger ϵ yields larger curvature-weighted variance proxies and typically larger entropy penalties, consistent with a tighter inferred posterior and weaker robust generalization.

This trend is expected because a larger ϵ expands the feasible set of adversarial perturbations and strengthens the inner

maximization. Training updates may then repeatedly accommodate harder, boundary-adjacent perturbations, which sustain sharp curvature modes in the robust loss for longer and amplify their magnitudes. In the dynamical PAC-Bayes instantiation, posterior contraction associated with curvature (through λ_i) competes with noise-driven inflation (through γ_i). Our ablations suggest that, as ϵ increases, the geometric tightening induced by sharper curvature is a dominant contributor to late-stage robust generalization degradation, even though the noise level also changes; this supports interpreting robust overfitting through posterior contraction when the curvature–noise balance becomes unfavorable after learning-rate drops.

Batch Size. In these experiments, batch size primarily changes the estimated gradient-noise level: larger batches reduce the stochasticity of SGD updates and shrink the projected noise variances. Figure 8 compares $B \in \{64, 128, 256\}$ (note that $B = 128$ matches our default AT setting in Figures 1 and 2). Across batch sizes, the curvature trajectories (λ) remain comparatively similar, while the projected noise variances (γ) shift by orders of magnitude: smaller batches produce larger γ_i, and larger batches suppress them. These shifts are reflected in the bound decomposition. Larger batches correspond to smaller inferred posterior variances σ_i^2 and therefore larger entropy proxies $-\sum \ln \sigma_i^2$, and they are accompanied by stronger late-stage degradation in robust test performance.

Under our stationary (and transient-window) approximations, the per-mode posterior variance scales with the projected noise level, $\sigma_i^2 \propto \gamma_i$ (modulated by learning rate, momentum, and curvature). Increasing B reduces γ_i, which contracts σ_i^2 and inflates the log-determinant (entropy) contribution to the KL term in the PAC-Bayes bound. This yields a dynamical view of a posterior-contraction diagnostic pattern: with insufficient noise injection (large B), the effective posterior volume shrinks rapidly, and the bound becomes dominated by KL/entropy penalties in sharp directions, especially after learning-rate decays. Smaller batches inject more noise, maintain larger posterior variances, and can delay entry into this collapse-dominated regime, thereby mitigating robust overfitting in the same curvature landscape.

Other Variations. Finally, we report extended diagnostics across architecture, learning rate schedules, objectives, and datasets, without altering the paper's central experimental design. Figures 9 and 10 use the same six-row grid format as the generalizability figures above, so each setting is shown with loss/error, curvature, noise, and bound-component diagnostics in one place. We use WRN-34-10 [Zagoruyko and Komodakis, 2016] to test a higher-capacity architecture than the default PreActResNet-18, TRADES [Zhang et al., 2019] to test an objective-level change to adversarial training, Imagenette-160 [Howard, 2019] to test a higher-resolution natural-image setting beyond CIFAR-scale inputs, and schedule variants to isolate learning-rate transitions. Unless otherwise stated, CIFAR-10 settings use ℓ_∞ perturbations at $\epsilon = 8/255$; Imagenette-160 uses ℓ_∞ perturbations at $\epsilon = 4/255$. The sharp two-drop schedule uses 0.1 for epochs 0–99, 0.001 for epochs 100–149, and 10^{-5} afterward. The one-drop schedule uses 0.1 before epoch 100 and 0.01 afterward.

These results show that the diagnostic pattern is not tied to the default PreActResNet-18/CIFAR-10/AT configuration. For WRN-34-10, the robust test loss increases after the learning-rate drop while the leading curvature and log-determinant diagnostics grow, consistent with the posterior-contraction pattern observed in the default AT run. The schedule variants isolate the role of learning-rate transitions: the sharp two-drop schedule produces abrupt changes in the variance and entropy diagnostics at the scheduled drops, whereas the one-drop schedule shows a milder post-drop increase in curvature and projected noise. The TRADES and Imagenette-160 runs preserve the same qualitative organization of the diagnostics: increases in leading curvature and projected noise are accompanied by larger curvature-weighted variance and log-determinant terms. These results support using the proposed diagnostic as a stress test across architecture, objective, dataset, and schedule choices, without changing the main claims or treating the top-k proxies as calibrated full-dimensional bounds.

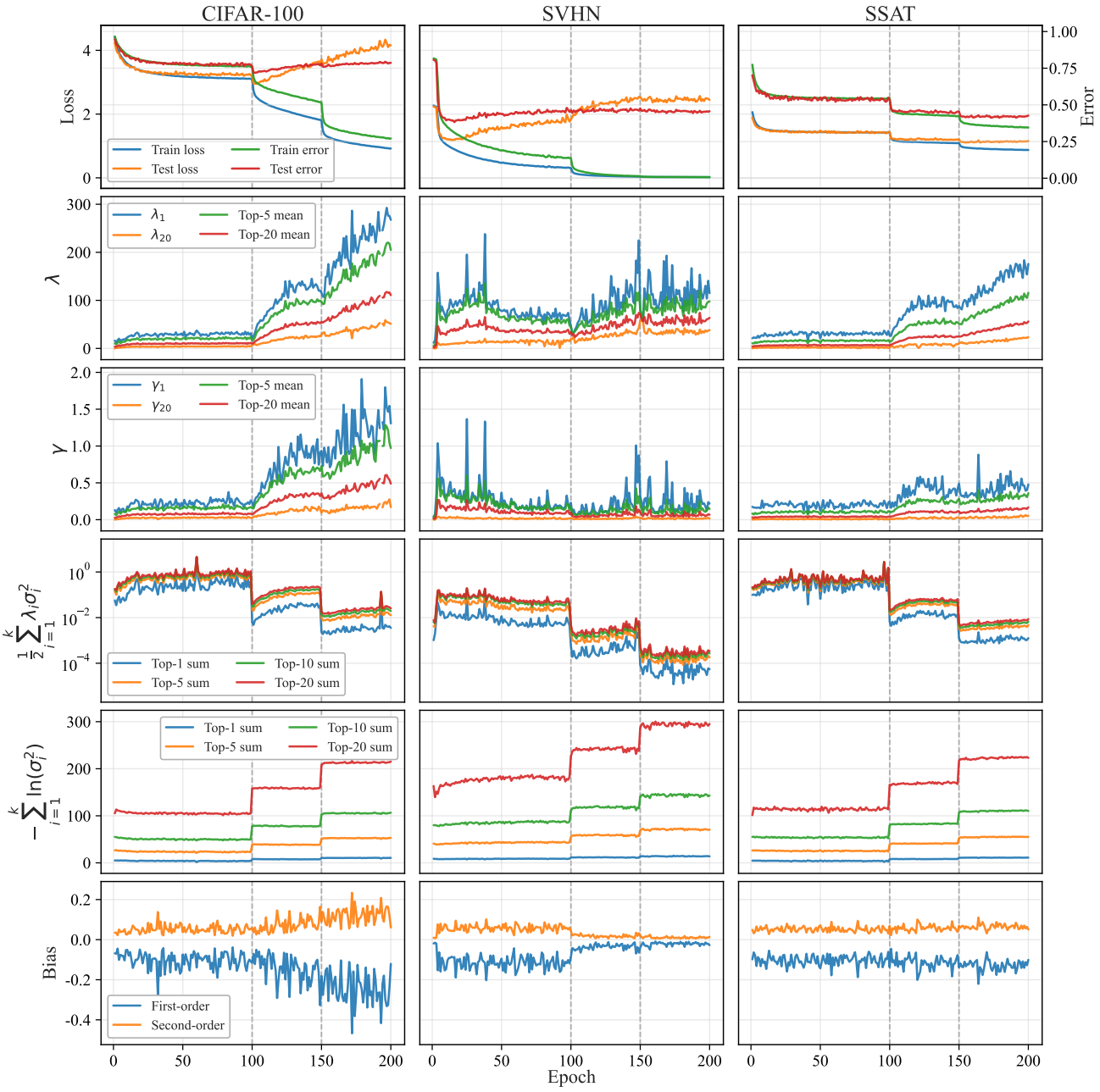

Figure 6: Diagnostic curves across datasets (CIFAR-100, SVHN) and learning algorithms (SSAT).

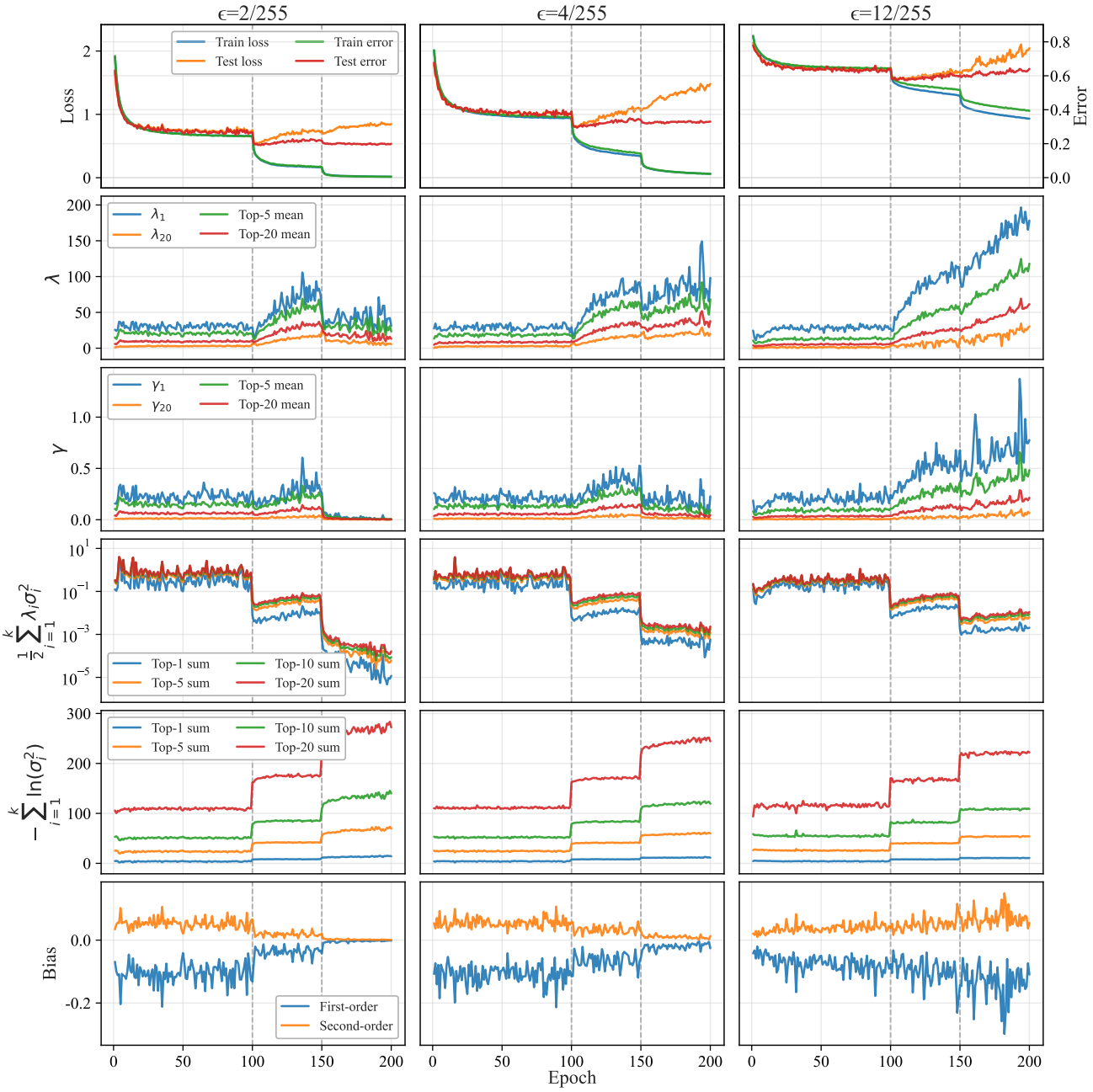

Figure 7: Diagnostic curves with varying perturbation radius $\epsilon \in \{2/255, 4/255, 12/255\}$.

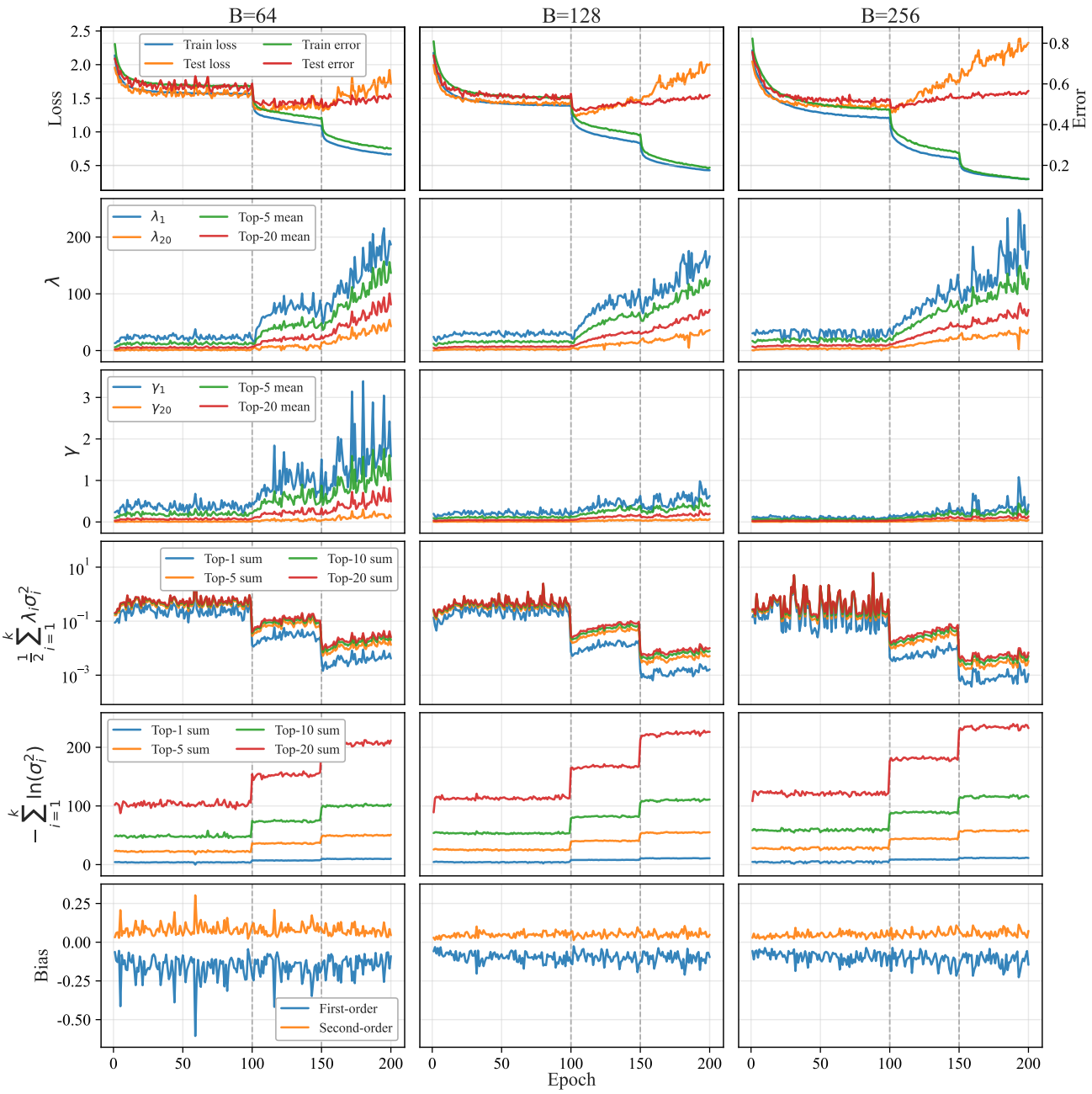

Figure 8: Diagnostic curves with varying batch size $B \in \{64, 128, 256\}$.

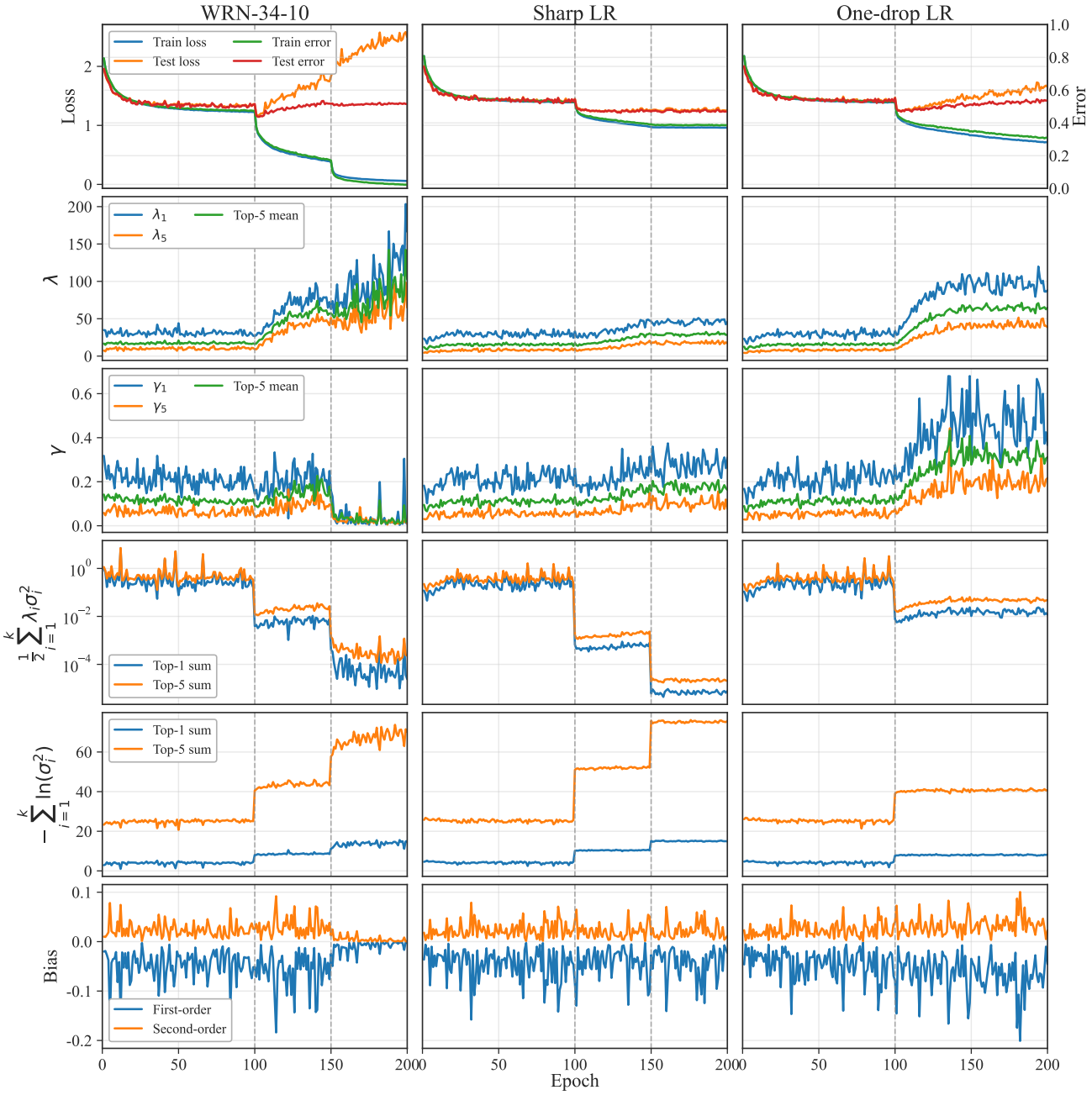

Figure 9: Diagnostic curves on WRN-34-10 and learning-rate schedules under ℓ_∞ perturbations with $\epsilon = 8/255$. Columns (from left to right) present the results on WRN-34-10, the sharp two-drop LR schedule, and the one-drop LR schedule.

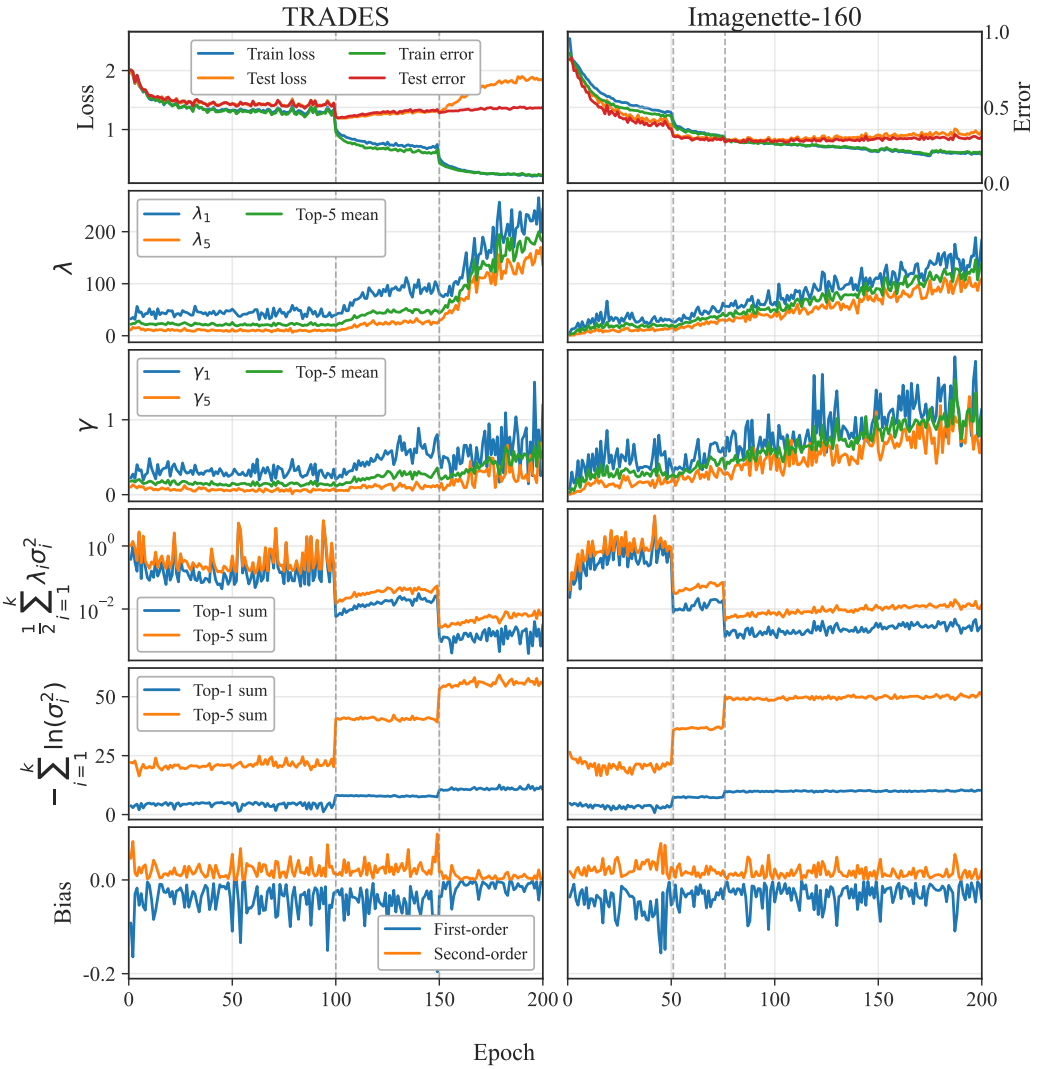

Figure 10: Extended diagnostics for objective-level and dataset-level variations. Columns refer to TRADES on CIFAR-10 under ℓ_∞ perturbations with $\epsilon = 8/255$ and AT on Imagenette-160 with ResNet-18 under ℓ_∞ perturbations with $\epsilon = 4/255$.