
A unified Bayesian framework for adversarial robustness

Pablo G. Arce^{1,2}

Roi Naveiro³

David Ríos Insua¹

¹Institute of Mathematical Sciences, Spanish National Research Council, Madrid, Spain

²Universidad Autónoma de Madrid, Escuela de Doctorado, Madrid, Spain

³CUNEF Universidad, Madrid, Spain

Abstract

The vulnerability of machine learning models to adversarial attacks remains a critical security challenge. Traditional defenses, such as adversarial training, typically robustify models by minimizing a worst-case loss. However, these deterministic approaches do not account for uncertainty in the adversary’s attack. While stochastic defenses placing a probability distribution on the adversary exist, they often lack statistical rigor and fail to make explicit their underlying assumptions. To resolve these issues, we introduce a formal Bayesian framework that models adversarial uncertainty through a stochastic channel, articulating all probabilistic assumptions. This yields two robustification strategies: a proactive defense enacted during training, aligned with adversarial training, and a reactive defense enacted during operations, aligned with adversarial purification. Several previous defenses can be recovered as limiting cases of our model. We empirically validate our methodology, showcasing the benefits of explicitly modeling adversarial uncertainty.

1 INTRODUCTION

The increasing importance of machine learning, amplified by large language models, underscores the transformative potential of AI (Zhao et al., 2023). However, this progress is shadowed by security issues, particularly the threat of adversarial attacks, which has given rise to the field of *adversarial machine learning* (AML) (Dalvi et al., 2004; Joseph et al., 2019). Adversaries break the core i.i.d. assumption by manipulating inputs, forcing the need for robust algorithms.

While AML is maturing for classical, point-estimate models, the adversarial robustness of Bayesian pre-

dictive models remains a critical and underexplored frontier (Feng et al., 2024). This is a significant gap, as Bayesian methods are essential in high-stakes domains, where principled uncertainty quantification is paramount. Existing work has mainly focused on demonstrating vulnerabilities of these models (Arce et al., 2025), but a principled foundation for designing defenses is still absent.

This paper establishes such foundation. We propose a fully Bayesian framework that models adversarial actions through a stochastic channel, allowing us to formally incorporate uncertainty about the adversary’s strategy. Our contributions include:

- A statistically grounded Bayesian framework for adversarial defenses that makes all assumptions transparent.
- The derivation of two strategies: a *reactive defense* for deployment and a *proactive defense* for training, along with tractable inference schemes.
- A demonstration that our framework generalizes prior art, recovering prominent defenses like *adversarial training* (AT) or *randomized smoothing* (RS) as limiting cases.

2 RELATED WORK

AML has gained significant attention as adversaries can manipulate data inputs to achieve malicious goals in critical settings (Joseph et al., 2019; Vorobeichyk and Kantarcioglu, 2019; Insua et al., 2023). While early AML work focused on classification (Goodfellow et al., 2014), the impact of these vulnerabilities is now recognized across diverse learning tasks, including regression (Arce et al., 2025) and reinforcement learning (Gallego et al., 2019). In general, defenses in AML fall into two categories: *proactive defenses* anticipating attacks during training, and *reactive defenses* acting on corrupted inputs during operations. However, most existing strategies suffer from two fundamental limitations: 1) they are essentially deterministic, failing

to quantify uncertainty about the adversary, and 2) they are designed for classical, point-estimate predictive models, not Bayesian ones.

The most prominent proactive defense is AT (Madry et al., 2018), which frames the problem as a mini-max optimization. An inner loop finds a worst-case attack, and an outer loop trains the model to minimize its loss on these attacks. This provides strong protection in many cases, but is doubly limited: its deterministic formulation ignores uncertainty in the adversary’s strategy, and it is inherently designed for point-estimate models, lacking a native mechanism to protect a full predictive distribution. Variants like TRADES (Zhang et al., 2019) or adversarial logit pairing (Kannan et al., 2018) decompose the robust loss into classification and regularization terms, but the core paradigm remains the same. Several heuristic variants have attempted to accommodate uncertainty, for instance through curriculum (Cai et al., 2018) or adaptive (Balaji et al., 2019) training.

Complementary to these are reactive defenses. The subfield of *adversarial purification* aims to remove perturbations from a corrupted input before classification, focusing on restoring a single “clean” input for a non-Bayesian model, rather than propagating the uncertainty about the original input through a Bayesian posterior. Among these strategies, we find *model-agnostic* ones, where the purifier is a generative model trained only on clean data, which purifies an input by projecting it back to the learned data manifold, independent of the downstream predictive model. A prime example uses diffusion models to iteratively denoise the input to find a likely clean predecessor (Nie et al., 2022). In contrast, *model-guided* strategies leverage the downstream predictive model parameters to actively shape the purification. Instead of just seeking a plausible clean instance, it seeks one that the specific predictive model is likely to predict correctly. For instance, *Atop* (Lin et al., 2024) uses the classifier gradients to guide a generative process, steering it towards high-confidence regions of the model decision boundary. Another key reactive strategy is RS (Cohen et al., 2019). Instead of deterministically purifying an input, it constructs a new, certifiably robust classifier by predicting the majority vote of a base classifier over several noisy versions of the input. While not Bayesian, RS is inherently probabilistic, as it smooths the decision boundary convolving it with a noise distribution. It serves as a crucial conceptual bridge towards uncertainty-aware defenses, though in its standard form, it does not involve posterior inference over model parameters.

While these classical paradigms are well-established, their limitations are particularly acute when facing

Bayesian models, whose attack surface is larger. Adversaries can target not only point predictions but also the entire posterior predictive distribution (Arce et al., 2025). This makes adversarial robustness of Bayesian models a critical and developing frontier. Despite early hopes that Bayesian methods might be inherently robust (De Palma et al., 2021), recent findings show this is not guaranteed, with attacks like PGD⁺ (Feng et al., 2024) or those in Arce et al. (2025) and (Carreau et al., 2025) demonstrating their vulnerabilities.

Prior attempts to create distinctly Bayesian defenses have faced their own challenges. Some are heuristic, such as considering distributions of attacks to inform a distributional AT (Dong et al., 2024). There are more formal ones like Bayesian Adversarial Learning (Ye and Zhu, 2018), which introduces a framework based on Gibbs sampling to approximate a robust posterior. However, as proved in Section 1 the Supplementary Material (SM), the conditional distributions defining their sampler are mathematically inconsistent and cannot be derived from any single, valid joint posterior distribution. More principled frameworks, like that of Gallego et al. (2024), have been limited in scope to classification problems only.

The absence of a framework that is both probabilistic in nature and natively designed for Bayesian models motivates this work. We introduce the first, to our knowledge, statistically rigorous and fully Bayesian framework that addresses these gaps. It not only models adversarial uncertainty via a stochastic channel but is specifically architected to robustify Bayesian predictive distributions. Furthermore, it recovers prominent defenses like AT and RS as limiting cases.

3 METHODOLOGY

3.1 Problem Formulation

We frame our problem within within the setting of Bayesian predictive models, where data $(\mathbf{x}, y)$ are drawn from a joint distribution $p(\mathbf{x}, y|\phi, \theta)$, factorized as $p(\mathbf{x}|\phi)p(y|\mathbf{x}, \theta)$, with parameters ϕ (for the covariates $\mathbf{x}$) and θ (for the labels y). Given a clean training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the standard objective is to learn the posterior $p(\phi, \theta|\mathcal{D})$ over these parameters and use it to form the posterior predictive distribution (PPD) $p(y|\mathbf{x}, \mathcal{D})$ for a new input $\mathbf{x}$.

The challenge we address arises at deployment. Under evasion attacks, we no longer observe the clean input $\mathbf{x}$. Instead, it is passed through an *adversarial channel* corrupting it to produce the observation $\mathbf{x}'$ in an attempt to confound the labeling process. To account for uncertainty in such corruption, we model this channel as a conditional distribution, $p(\mathbf{x}'|\mathbf{x}, \theta)$, allowing

its form to depend on the model parameters reflecting an adversary with potential access to the system. The central problem is therefore to provide a reliable prediction for y given only the corrupted $\mathbf{x}'$.

To solve this, we propose two strategies. The first one is a *reactive defense*, designed to protect the model during operations. It uses a standardly trained model but, upon receiving a possibly corrupted input at test time, employs a robust inference mechanism to account for the channel, in the spirit of *adversarial purification* methods in AML. The second strategy is a *proactive defense* that builds robustness directly within the training phase. By contemplating the adversarial channel into the learning objective, this method yields a novel Bayesian formulation of the classic AT paradigm.

The following subsections develop these formal models and their inference and prediction schemes. They rely on two different graphical models presented in Figures 1 and 2. While addressing the same challenge, these two strategies are mathematically distinct and lead to different PPDs, a claim formally proved in Section 2 of the SM. In their conception, the definition of the adversarial channel is a crucial and flexible component of our framework, accommodating different assumptions about the adversary. In its simplest form, the channel can be an attack-agnostic model, like isotropic Gaussian noise, which connects our framework to defenses like RS (see Section 3 of the SM). More powerfully, it can be an attack-based channel, taking a standard deterministic attack, such as projected gradient descent (PGD), and making it probabilistic by placing priors over its parameters or injecting noise into its outputs. To model a more sophisticated adversary, this can be extended to a mixture channel, where each component is itself a full probabilistic attack-based channel (e.g., one for Carlini & Wagner (CW), another for PGD), each with priors. Finally, the channel can be a learned generative model, where a separate neural network (NN) is trained to produce the attack distribution. Our experiments will explore both attack-based and learned channels to demonstrate this versatility.

3.2 Protection During Operations

A natural approach to defending a model during deployment is to assume that the labeling mechanism is invariant under attack: while an adversary may corrupt an input from $\mathbf{x}$ to $\mathbf{x}'$, the label y remains conditionally dependent only on the parameters θ and the original, now latent, covariate vector $\mathbf{x}$. This is captured through the probabilistic graphical model in Figure 1, depicting a standard training phase on clean data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ used to learn the posterior over ϕ and θ . At test time, the defense is enacted upon ob-

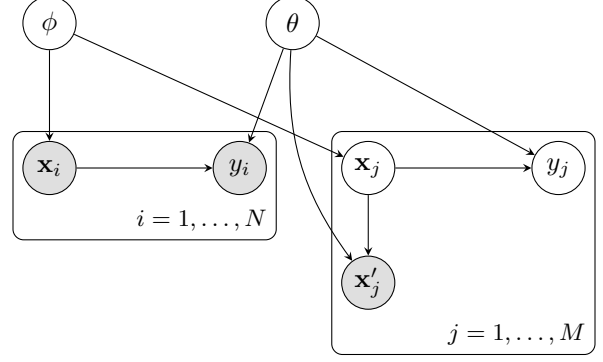


Figure 1: Model for reactive defense

serving a possibly corrupted input $\mathbf{x}'_j$; it reasons backward through the adversarial channel to infer the latent clean input $\mathbf{x}_j$ and thereby predict the label y_j .

Therefore, this reactive defense computes the robust predictive distribution $p(y_j|\mathbf{x}'_j, \mathcal{D})$. A full Bayesian treatment requires marginalizing over all unobserved quantities, as shown in the integral,

$$p(y_j|\mathbf{x}'_j, \mathcal{D}) = \iiint p(y_j|\mathbf{x}_j, \theta) p(\mathbf{x}_j, \theta, \phi|\mathbf{x}'_j, \mathcal{D}) d\mathbf{x}_j d\theta d\phi, \quad (1)$$

which depends on the joint posterior $p(\mathbf{x}_j, \theta, \phi|\mathbf{x}'_j, \mathcal{D})$, hence coupling the latent variable $\mathbf{x}_j$ with the global parameters (θ, ϕ) and the full training dataset $\mathcal{D}$.

Attempting to approximate (1) directly, for instance by sampling, is often intractable: first, the joint space of covariates $\mathbf{x}_j$ and parameters (θ, ϕ) is typically high-dimensional, posing a significant challenge for MCMC methods; second, if the stochastic adversarial channel $p(\mathbf{x}'_j|\mathbf{x}_j, \theta)$ is only accessible via sampling (as with a black-box simulator), the problem becomes one of likelihood-free inference, introducing further complexity. To develop a practical approach, we introduce a key simplifying assumption: test points $\mathbf{x}'_j$ provide negligible information about the global parameters, suggesting the approximation $p(\theta, \phi|\mathbf{x}'_j, \mathcal{D}) \approx p(\theta, \phi|\mathcal{D})$. We refer to the defense based on this simplification as *offline reactive*, as it relies on a fixed posterior learned offline. This decouples inference on parameters from inference on the latent input, allowing us to write the PPD (1) as a nested expectation

$$p(y_j|\mathbf{x}'_j, \mathcal{D}) = \mathbb{E}_{(\theta, \phi)|\mathcal{D}} \left[\mathbb{E}_{\mathbf{x}_j|\mathbf{x}'_j, \theta, \phi} [p(y_j|\mathbf{x}_j, \theta)] \right]. \quad (2)$$

Under this assumption, inference on (θ, ϕ) can be performed off-line using standard methods. The remaining challenge is the on-line computation of the inner expectation in (12), which requires inferring the latent covariate vector $\mathbf{x}_j$ given the observed attack $\mathbf{x}'_j$. This online inference problem is akin to the central task

of *adversarial purification*. Most purification methods can be viewed as non-Bayesian counterparts to our approach. Instead of working with the full posterior $p(\mathbf{x}_j|\mathbf{x}'_j, \theta, \phi)$, they approximate it with a point mass at a single, restored estimate $\hat{\mathbf{x}}_j$. These methods are typically *model-agnostic*, if this purification is independent of the downstream predictive model parameters θ , or *model-guided*, if θ is used to shape the restoration process. Beyond adversarial purification, our framework also encompasses other defenses such as RS (Cohen et al., 2019), a leading method for certifiable adversarial defense, which emerges as a special case by defining a simple noise model, as Section 3 of the SM proves.

Notice that the inner expectation (12) can be expanded using Bayes' rule on $p(\mathbf{x}_j|\mathbf{x}'_j, \theta, \phi)$ yielding

$$\mathbb{E}_{\theta, \phi|\mathcal{D}} \left[\mathbb{E}_{\mathbf{x}_j|\phi} \left[\frac{p(\mathbf{x}'_j|\mathbf{x}_j, \theta)p(y_j|\mathbf{x}_j, \theta)}{\mathbb{E}_{\mathbf{x}_j|\phi} [p(\mathbf{x}'_j|\mathbf{x}_j, \theta)]} \right] \right],$$

where we use that $\mathbf{x}'_j$ is conditionally independent from ϕ given θ and $\mathbf{x}'_j$, and $\mathbf{x}_j$ is conditionally independent from θ given ϕ . This reveals that calculating the predictive distribution for y_j relies on computing the generative model $p(\mathbf{x}_j|\phi)$ and the channel normalizing constant $\mathbb{E}_{\mathbf{x}_j|\phi} [p(\mathbf{x}'_j|\mathbf{x}_j, \theta)]$.

A straightforward, non-parametric approach to bypass the challenge of learning an explicit deep generative model is to use the empirical distribution of the training data as a substitute. This leads to a simple and intuitive algorithm where the predictive distribution is approximated as a weighted sum over posterior samples and the training data points

$$p(y_j|\mathbf{x}'_j, \mathcal{D}) \approx \frac{1}{S} \sum_{s=1}^S \sum_{i=1}^N w_{si} p(y_j|\mathbf{x}_i, \theta_s), \quad (3)$$

where $\{\theta_s\}_{s=1}^S$ are samples from the posterior $p(\theta|\mathcal{D})$ with importance weights $w_{si} = \frac{p(\mathbf{x}'_j|\mathbf{x}_i, \theta_s)}{\sum_{k=1}^N p(\mathbf{x}'_j|\mathbf{x}_k, \theta_s)}$. However, while conceptually simple, this approach faces severe practical limitations. First, it has tremendous memory costs, as it requires the entire training dataset to be stored and accessible at test time, making it inefficient in realistic, large-scale scenarios. Second, a more fundamental challenge is the need to evaluate the adversarial channel probability, $p(\mathbf{x}'_j|\mathbf{x}_i, \theta_s)$. For many realistic attacks, such as those based on iterative optimization, this probability is not available in a closed form. This latter issue recasts the problem of inferring the latent $\mathbf{x}_j$ as one of likelihood-free inference, as we can simulate from the channel but cannot evaluate its likelihood. While methods like Approximate Bayesian Computation have been proposed to solve this (Gallego et al., 2024), they scale poorly with

the dimensionality of the covariate space. More advanced strategies like sequential neural posterior estimation (Papamakarios et al., 2019) are promising, but fall outside the scope of this paper.

As an alternative to the offline approximation, we can derive an online adaptive defense by avoiding the assumption $p(\theta, \phi|\mathbf{x}'_j, \mathcal{D}) \approx p(\theta, \phi|\mathcal{D})$. As proven in Section 4 of the SM, in this case $p(y_j|\mathbf{x}'_j, \mathcal{D})$ is

$$\frac{\mathbb{E}_{\theta, \phi|\mathcal{D}_{\text{train}}} [\mathbb{E}_{\mathbf{x}_j|\phi} [p(y_j|\mathbf{x}_j, \theta)p(\mathbf{x}'_j|\mathbf{x}_j, \theta)]]}{\mathbb{E}_{\theta, \phi|\mathcal{D}_{\text{train}}} [\mathbb{E}_{\mathbf{x}_j|\phi} [p(\mathbf{x}'_j|\mathbf{x}_j, \theta)]]}.$$

Then, using the empirical distribution as the generative model of covariates, we get

$$p(y_j|\mathbf{x}'_j, \mathcal{D}) \approx \frac{\sum_{s=1}^S \sum_{i=1}^N p(y_j|\mathbf{x}_i, \theta_s)p(\mathbf{x}'_j|\mathbf{x}_i, \theta_s)}{\sum_{s'=1}^S \sum_{k=1}^N p(\mathbf{x}'_j|\mathbf{x}_k, \theta_{s'})}. \quad (4)$$

The difference between this and the offline defense is how predictions from each posterior sample θ_s are aggregated. The offline defense treats each posterior sample as equally likely, applying a uniform weight $1/S$ to its purified prediction. The online defense uses the new $\mathbf{x}'_j$ to compute non-uniform weights, effectively performing a Bayesian update at test time. As Section 4 of the SM shows, this is equivalent to reweighting the prediction from each posterior sample θ_s by its marginal likelihood for the observed attack $\sum_{i=1}^N p(\mathbf{x}'_j|\mathbf{x}_i, \theta_s)$, making the online defense more adaptive, as it uses the new evidence $\mathbf{x}'_j$ to give more influence to parameters that better explain the attack.

Anyway, both approaches share the same fundamental limitations, namely the high online computational costs, memory requirements, and the difficulty of evaluating the adversarial channel. We therefore turn to proactive defenses in the next section.

3.3 Protection During Training

Consider now proactively training the model to be inherently robust, shifting the computational effort from the test to an offline training phase. For this, we alter the assumed generative process, introducing a latent, fictitious adversarial example $\mathbf{x}'_i$ for each training point, as Figure 2 shows. The label y_i is now assumed to be generated from this unobserved corrupted input. This proactive approach fundamentally changes the inference problem, resolving the main computational challenges of the reactive defense.

A full Bayesian treatment of this model requires marginalizing out the latent variable $\mathbf{x}'_i$ in the likelihood calculation. The likelihood for a single observation $(\mathbf{x}_i, y_i)$ is factorized as

$$p(\mathbf{x}_i, y_i|\theta, \phi) = p(y_i|\mathbf{x}_i, \theta, \phi) p(\mathbf{x}_i|\phi)$$

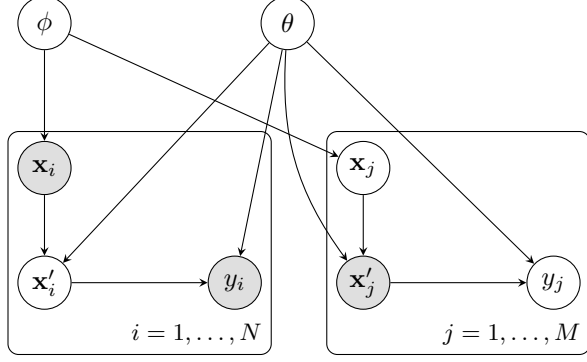


Figure 2: Model for proactive defense, where the training process explicitly models the adversarial channel.

If we marginalize out the latent adversarial example $\mathbf{x}'_i$ in the likelihood of the label y_i through

$$p(y_i|\mathbf{x}_i, \theta, \phi) = \int p(y_i|\mathbf{x}_i, \theta) p(\mathbf{x}'_i|\mathbf{x}_i, \theta) d\mathbf{x}'_i, \quad (5)$$

where we use that 1) y_i is conditionally independent of all other nodes given $\mathbf{x}'_i$ and θ , and 2) $\mathbf{x}'_i$ is conditionally independent of ϕ given $\mathbf{x}_i$ and θ . Then, it is straightforward to see that the full joint posterior $p(\theta, \phi|\mathcal{D})$ factorizes, being proportional to,

$$\left[p(\theta) \prod_{i=1}^N \mathbb{E}_{\mathbf{x}'_i|\mathbf{x}_i, \theta} (p(y_i|\mathbf{x}'_i, \theta)) \right] \cdot \left[p(\phi) \prod_{i=1}^N p(\mathbf{x}_i|\phi) \right], \quad (6)$$

provided we choose independent priors for θ and ϕ . This derivation reveals a key advantage of the proactive approach: to learn the robust predictive model, we no longer need to perform joint inference or specify a generative model $p(\mathbf{x}|\phi)$.

This principled formulation generalizes standard AT, which can be recovered as a deterministic limit of our model by using a point mass for the adversarial channel and a point estimate for the parameters θ , as Section 5 of the SM shows. Crucially, this generalization is not merely theoretical. By replacing the deterministic worst-case adversary with a stochastic channel, our framework naturally models uncertainty about the attacker’s strategy, moving beyond defending against a single, specific threat model. As our experiments demonstrate, training against a distribution of attacks can confer robustness against entirely different attack modalities not seen during training, a significant advantage over methods tuned to a specific worst-case adversary as with AT.

A second advantage of this approach becomes evident at prediction time. Having performed the computationally intensive robust training offline, making a prediction for a new adversarial input $\mathbf{x}'_j$ is efficient. The

predictive distribution follows the standard Bayesian process of marginalizing over the learned posterior

$$\begin{aligned} p(y_j|\mathbf{x}'_j, \mathcal{D}) &= \iint p(y_j|\mathbf{x}'_j, \theta, \phi) p(\theta, \phi|\mathbf{x}'_j, \mathcal{D}) d\theta d\phi \\ &= \int p(y_j|\mathbf{x}'_j, \theta) p(\theta|\mathbf{x}'_j, \mathcal{D}) d\theta. \end{aligned}$$

Due to the model structure (Figure 2), label y_j is conditionally independent of the covariate parameters ϕ given θ , allowing ϕ to be marginalized out. In addition, if, as before, we assume that $p(\theta|\mathbf{x}'_j, \mathcal{D}) \approx p(\theta|\mathcal{D})$, the final expression is the standard posterior predictive distribution, although taken over the posterior computed with the likelihood (5) which we refer to as robust posterior. Consequently, no online purification, access to the training data, or generative model is required at test time. The entire defense mechanism is encapsulated within the robust posterior $p(\theta|\mathcal{D})$, making prediction as fast as with a standard non-robust Bayesian model.

While the proactive defense simplifies the predictive task, its training phase remains challenging because the robust posterior in (6) is computationally intractable. The likelihood for each data point is itself an integral, which precludes a closed-form solution. To make this approach practical, we propose using variational inference (VI) (Blei et al., 2017) to approximate the true posterior with a tractable, parameterized distribution $q_\psi(\theta)$. This is achieved by maximizing the Evidence Lower Bound (ELBO)

$$\begin{aligned} \mathcal{L}(\psi) &= \mathbb{E}_{\theta \sim q_\psi(\theta)} \left[\sum_{i=1}^N \log \mathbb{E}_{\mathbf{x}'_i \sim p(\cdot|\mathbf{x}_i, \theta)} [p(y_i|\mathbf{x}'_i, \theta)] + \right. \\ &\quad \left. \log p(\theta) - \log q_\psi(\theta) \right]. \end{aligned} \quad (7)$$

Optimizing this ELBO with standard stochastic gradient methods is difficult due to the log of an expectation term within the sum. To bypass this issue, we apply Jensen’s inequality to the inner expectation in (7). This yields a tractable lower bound on the ELBO, $\mathcal{L}(\psi) \geq \tilde{\mathcal{L}}(\psi)$. We then maximize this new, more manageable objective $\tilde{\mathcal{L}}(\psi)$, given by

$$\begin{aligned} &\mathbb{E}_{\theta \sim q_\psi(\theta)} \left[\sum_{i=1}^N \mathbb{E}_{\mathbf{x}'_i \sim p(\cdot|\mathbf{x}_i, \theta)} [\log p(y_i|\mathbf{x}'_i, \theta)] \right] \\ &- \text{KL}(q_\psi(\theta) \| p(\theta)). \end{aligned}$$

This objective is a double expectation, fully amenable to stochastic optimization. When the variational posterior is reparameterizable, $\theta = f(\psi, \epsilon)$ with $\epsilon \sim p(\epsilon)$, and the adversarial channel is also reparameteriz-

able¹ $\mathbf{x}' = h(\mathbf{x}, \theta, \boldsymbol{\eta})$, with $\boldsymbol{\eta} \sim p(\boldsymbol{\eta})$, the gradient $\nabla_{\psi} \tilde{\mathcal{L}}(\psi)$ can be moved inside both expectations of $\tilde{\mathcal{L}}(\psi)$ and takes the form

$$\mathbb{E}_{\epsilon} \left[\sum_{i=1}^N \mathbb{E}_{\boldsymbol{\eta}} \nabla_{\psi} \log p(y_i | \mathbf{x}'_i(\boldsymbol{\eta}), \theta(\psi, \epsilon)) \right] - \nabla_{\psi} KL(q_{\psi}(\theta) \| p(\theta)).$$

The first term is now amenable to unbiased estimation by MC, while the second will have an exact expression for some variational families. This provides an efficient method for learning the robust posterior. An alternative strategy which estimates the gradient of the original ELBO is detailed in Section 6 of the SM.

4 EXPERIMENTS

We conduct a series of experiments to demonstrate the empirical advantages of our proposed Bayesian defense framework. For the computational reasons outlined in Section 3.2, our evaluation centers on proactive defenses, providing a comprehensive robustness analysis against strong baselines, although we also offer a conceptual validation of the reactive approach.

4.1 Experimental Setup

We evaluate our framework across both classification and regression tasks. For image classification, we use the MNIST dataset (LeCun et al., 1998), whereas for regression we consider the Wine (Cortez et al., 2009) and Energy Efficiency (Tsanas and Xifara, 2012) datasets. The underlying predictive model is always a Bayesian NN (BNN) trained with VI. For classification, we employ a convolutional architecture, whereas for regression we use a fully connected architecture. Code to reproduce the experiments as well as a full parameter specification can be found at <https://anonymous.4open.science/r/advDef>.

We benchmark our approach against standard AT, implemented by augmenting each minibatch with its adversarially perturbed counterparts, allowing the model to learn from both clean and attacked inputs. As a baseline, we also consider the undefended BNN trained using the standard ELBO objective on clean data. For robustness assessment we use several adversarial attacks bounded in L_2 norm by ϵ : a single-step PGD attack (PGD1), a multi-step PGD with 50 iterations (PGD), an entropy-based PGD variant whose objective is to maximize predictive entropy (ENT), and the PGD⁺ attack by Feng et al. (2024), which undertakes

25 iterations of PGD and 25 iterations of entropy-based PGD. Performance is evaluated using metrics that measure both deterministic accuracy and probabilistic quality. We report accuracies for classification and RMSE for regression. In addition, we evaluate the negative log-likelihood (NLL) for both settings, complementing evaluation with a proper scoring rule that accounts for predictive uncertainty and calibration. Experiments are conducted on three test sets. Tables report means with standard deviations in parentheses, while Figures display means with ± 1 standard deviation bands. Approximate inference is performed using stochastic VI with the Adam optimizer and minibatching. Further experimental details and additional results are provided in Section 7 of the SM. Unless otherwise specified, default hyperparameters are used, with complete specifications detailed in the repository. All experiments are executed on a computing node with three NVIDIA A100 GPUs.

In our experiments, we instantiate the adversarial channels from our framework into several proactive training regimes. We explore attack-based channels built from a one-step PGD attack, made probabilistic by perturbing the attacked input with Gaussian noise. This yields two models: OS, trained exclusively on these attacks, and OS50, trained on minibatches of 50% attacked and 50% clean data. Second, our MIX model implements a mixture channel, which combines several different probabilistic attack types (PGD, PGD+, CW, etc.) and attack parameters to simulate a more diverse adversary. Third, we instantiate a learned generative channel in our NN and NN50 models, where a separate NN generates the attack distribution; as before, NN50 uses a 50/50 training mix while NN is trained solely on attacks. We benchmark these defenses against a baseline undefended BNN trained on clean data (BL) and a conventional AT implementation, also trained on a 50/50 mix of clean and one-step attacked inputs. Finally, we also assess our two reactive defenses, the offline (offPure) and the online adaptive (onPure) models, corresponding to equations (3) and (4), respectively.

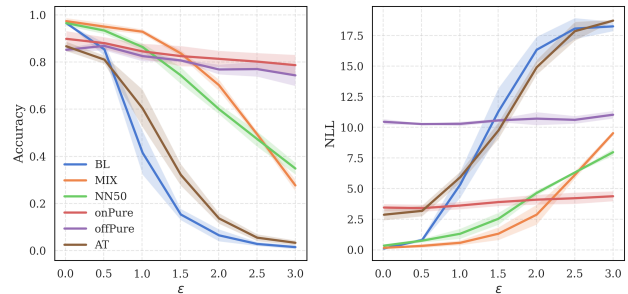
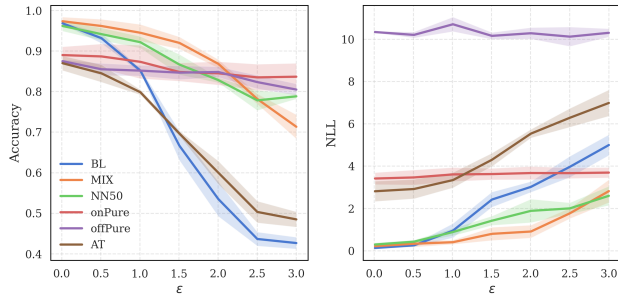


Figure 3: Accuracy and NLL against PGD50 attack.

¹This condition is met by the attack-based channels we consider, as they introduce randomness to a deterministic attack in a reparameterizable fashion.


Figure 4: Accuracy and NLL against PGD⁺ attack.

4.2 Case Study: Classification

Evaluating Defenses We evaluate our proposed methodology against strong white-box attacks on MNIST. Figures 3 and 4 present performance metrics against attack intensity for PGD and the multi-objective PGD⁺, respectively.

As anticipated, the undefended baseline exhibits rapid degradation under attack. AT provides moderate robustness improvements but suffers from poor clean accuracy while maintaining elevated NLL values, indicating limited benefits. The purification methods reveal distinct trade-offs compared to AT. offPure consistently exhibits very high NLL values across all perturbation strengths. While onPure limits this and achieves substantially lower NLLs, both approaches degrade clean accuracy due to oversmoothing of the predictive distribution. This phenomenon, detailed in Section 3 of the SM, stems from the relationship between (oversimplified) purification and RS.

In contrast, MIX and NN50 retain clean accuracy while achieving competitive robustness. MIX benefits from exposure to diverse adversaries during training but requires prior assumptions about the attack space. Most notably, NN50—trained exclusively against a learned NN adversary without assuming any fixed attack strategy—demonstrates consistently strong performance across all metrics. This specification-free approach proves highly effective, suggesting that learned adversarial methods can achieve defenses similar to traditional AT approaches.

Downstream Task: Selective Accuracy To assess the robustness of our defenses against attacks targeting uncertainty, we evaluate performance on a selective prediction task. This experiment simulates an uncertainty-based filtering approach where the predictive entropy of the trained BNN serves as an out-of-distribution detector. We construct a balanced test set of MNIST and FashionMNIST samples and attack MNIST samples with PGD⁺ to simultaneously induce misclassification and increase entropy and FashionM-

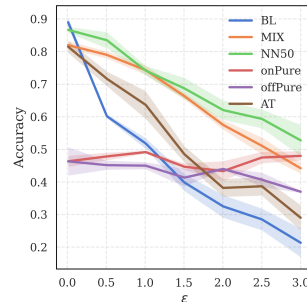


Figure 5: Selective accuracy.

NIST samples with an entropy attack to decrease entropy and evade detection. We measure accuracy on the retained samples after filtering out the half with highest uncertainty scores.

Figure 5 presents results across varying attack strengths. The findings largely mirror our previous observations, with two notable distinctions. First, both purification-based models exhibit reduced clean accuracy, though onPure demonstrates a marginal advantage. This performance degradation can be attributed to the oversimplifications made in the reactive approach. Second, although reactive approaches previously showed advantages over NN50 and MIX under stronger attacks, the latter achieve superior performance in terms of selective accuracy for all intensities.

Table 1: Accuracies at $\epsilon = 2$.

Model	Clean	PGD1	PGD	PGD ⁺	ENT
BL	0.96 (0.00)	0.32 (0.03)	0.06 (0.03)	0.53 (0.04)	0.89 (0.00)
OS	0.84 (0.01)	0.72 (0.02)	0.58 (0.01)	0.73 (0.01)	0.83 (0.02)
OS50	0.96 (0.01)	0.81 (0.02)	0.67 (0.06)	0.84 (0.03)	0.93 (0.02)
MIX	0.97 (0.01)	0.86 (0.03)	0.70 (0.02)	0.87 (0.01)	0.94 (0.02)
NN	0.59 (0.02)	0.54 (0.03)	0.47 (0.01)	0.55 (0.03)	0.58 (0.01)
NN50	0.96 (0.01)	0.81 (0.01)	0.60 (0.02)	0.83 (0.01)	0.93 (0.01)
onPure	0.89 (0.03)	0.83 (0.04)	0.81 (0.03)	0.84 (0.03)	0.88 (0.05)
offPure	0.87 (0.01)	0.80 (0.01)	0.77 (0.02)	0.85 (0.01)	0.86 (0.02)
AT	0.87 (0.02)	0.44 (0.03)	0.14 (0.02)	0.60 (0.03)	0.79 (0.02)

Table 2: NLLs at $\epsilon = 2$.

Model	Clean	PGD1	PGD	PGD ⁺	ENT
BL	0.12 (0.04)	6.95 (0.74)	16.33 (1.06)	3.02 (0.25)	0.41 (0.10)
OS	2.90 (0.26)	3.89 (0.23)	5.62 (0.45)	3.91 (0.13)	3.09 (0.33)
OS50	0.37 (0.12)	1.70 (0.29)	3.56 (0.97)	1.65 (0.43)	0.81 (0.09)
MIX	0.22 (0.09)	1.25 (0.19)	2.88 (0.81)	0.91 (0.30)	0.43 (0.15)
NN	9.24 (0.34)	9.83 (0.56)	10.70 (0.54)	9.72 (0.52)	9.37 (0.50)
NN50	0.38 (0.06)	1.91 (0.30)	4.64 (0.22)	1.89 (0.55)	0.81 (0.09)
onPure	3.46 (0.34)	3.79 (0.31)	4.10 (0.40)	3.68 (0.31)	3.38 (0.29)
offPure	10.18 (0.24)	10.78 (0.28)	10.71 (0.53)	10.28 (0.27)	9.69 (0.16)
AT	2.82 (0.47)	7.68 (0.34)	14.90 (0.61)	5.54 (0.20)	3.24 (0.56)

Ablation Study We conduct an ablation study to isolate the sources of robustness in our proactive defense, investigating two central components: balanced training and the stochastic channel. The findings, summarized in Tables 1 and 2, reveal the distinct impact of each. Models trained exclusively on adversarial examples (OS, NN) suffer significant degra-

Table 3: RMSEs on Wine and Energy datasets at $\epsilon = 2$ under different attack types.

Model	Wine					Energy				
	Clean	PGD1	PGD	PGD ⁺	ENT	Clean	PGD1	PGD	PGD ⁺	ENT
BL	0.75 (0.04)	15.32 (0.05)	15.19 (0.00)	22.47 (0.14)	15.63 (0.03)	2.32 (1.34)	8.14 (0.15)	8.55 (0.21)	9.73 (0.12)	8.11 (0.36)
MIX	0.74 (0.02)	1.11 (0.00)	1.09 (0.01)	1.53 (0.01)	1.12 (0.03)	2.27 (1.01)	2.95 (0.18)	2.93 (0.10)	4.69 (0.21)	2.95 (0.29)
NN50	0.79 (0.06)	0.93 (0.06)	0.90 (0.06)	0.96 (0.06)	0.89 (0.06)	2.25 (0.79)	3.86 (0.16)	3.71 (0.16)	5.91 (0.31)	3.86 (0.17)
onPure	1.11 (0.03)	1.07 (0.02)	1.11 (0.01)	1.08 (0.03)	1.08 (0.01)	2.96 (0.13)	3.03 (0.30)	3.19 (0.38)	3.17 (0.12)	3.10 (0.24)
offPure	1.13 (0.06)	1.13 (0.08)	1.15 (0.04)	1.14 (0.02)	1.10 (0.03)	2.85 (0.21)	2.88 (0.26)	2.98 (0.38)	3.18 (0.15)	3.13 (0.33)
AT	0.76 (0.02)	6.75 (0.03)	6.46 (0.09)	8.60 (0.05)	6.70 (0.02)	2.41 (1.09)	6.25 (0.34)	6.26 (0.24)	8.59 (0.36)	6.26 (0.49)

Table 4: NLLs on Wine and Energy datasets at $\epsilon = 2$ under different attack types.

Model	Wine					Energy				
	Clean	PGD1	PGD	PGD ⁺	ENT	Clean	PGD1	PGD	PGD ⁺	ENT
BL	1.16 (0.06)	173.51 (1.97)	149.05 (0.95)	316.88 (3.57)	185.29 (1.23)	0.64 (0.03)	172.29 (4.56)	160.89 (7.53)	221.56 (0.98)	168.03 (4.11)
MIX	1.12 (0.02)	1.72 (0.03)	1.77 (0.07)	2.56 (0.08)	1.69 (0.04)	1.50 (0.07)	3.47 (0.28)	3.33 (0.15)	6.78 (0.68)	3.49 (0.34)
NN50	1.22 (0.05)	1.36 (0.05)	1.34 (0.06)	1.40 (0.06)	1.34 (0.05)	1.75 (0.05)	3.52 (0.19)	3.48 (0.21)	5.59 (0.33)	3.65 (0.20)
onPure	1.25 (0.06)	1.30 (0.06)	1.35 (0.08)	1.42 (0.07)	1.32 (0.06)	6.09 (0.63)	7.41 (0.85)	7.23 (0.74)	7.97 (0.86)	7.28 (0.68)
offPure	6.69 (0.33)	6.94 (0.38)	7.04 (0.51)	7.48 (0.49)	6.99 (0.57)	30.52 (3.14)	36.98 (4.21)	36.11 (3.69)	39.66 (4.20)	36.32 (3.44)
AT	1.15 (0.01)	21.70 (0.67)	22.88 (0.74)	30.91 (0.71)	20.43 (0.68)	1.33 (0.05)	14.00 (1.25)	13.09 (1.60)	23.19 (1.09)	14.78 (2.13)

dation in performance on clean data. In contrast, balanced approaches (OS50, NN50) that train on a 50/50 mix of clean and attacked inputs achieve strong clean accuracy (>0.96) and well-calibrated predictions while maintaining competitive robustness. The benefit of the stochastic channel is even more pronounced: the deterministic AT baseline, despite using balanced training, is significantly outperformed by its stochastic equivalent, OS50, across all metrics. This demonstrates that our probabilistic formulation enables more effective learning from adversarial examples than balanced training alone can provide. The benefit of adversarial diversity is further validated by MIX and NN50.

4.3 Case Study: Regression

To demonstrate our approach’s broad applicability, we extend our evaluation to regression tasks using the Wine and Energy datasets. Tables 3 and 4 confirm our methodology generalizes effectively beyond classification. The undefended baseline exhibits catastrophic failure under attack for both datasets, with RMSE increasing dramatically (e.g. 0.75 to over 15 on Wine) and NLL values exploding, indicating complete loss of predictive reliability. On the Wine dataset, NN50 achieves the best overall performance, maintaining reasonable clean accuracy with modest degradation under attack (RMSE from 0.79 to 0.96 under the worst attack) and well-calibrated predictions (NLL around 1.4). MIX performs competitively, achieving the lowest clean RMSE (0.74) but showing slightly higher attack sensitivity. On Energy dataset, MIX and NN50 perform similarly, with MIX slightly stronger in RMSE and both comparable in NLL.

The purification methods retain their trade-offs: while both yield RMSE competitive with the best proactive defenses, onPure provides reasonable calibration

whereas offPure suffers from severe calibration issues. These findings confirm that our key insights regarding balanced adversarial training and purification limitations hold consistently across learning paradigms.

5 CONCLUSIONS

We have introduced a statistically rigorous and fully Bayesian framework for adversarial defense, addressing a critical gap in the robustness of Bayesian predictive models. By modeling the adversary’s actions through a stochastic channel, our framework makes all probabilistic assumptions transparent. It yields two complementary strategies: a *reactive defense* that provides a principled foundation for adversarial purification, and a *proactive defense* that generalizes AT by incorporating uncertainty about the attack. We formally prove that prominent defenses like AT and RS are recovered as limiting cases of our framework. Our empirical results validate the proactive approach, showcasing that explicitly modeling adversarial uncertainty confers superior robustness in both model accuracy and the quality of predictive distributions.

This work opens avenues for future research. For reactive defenses, the primary challenge is computational efficiency; promising directions include exploring likelihood-free inference methods to make the more adaptive “online” version of our model practical. For proactive defenses, future work could involve developing more sophisticated learned adversarial channels, for instance by training an amortized generative adversary. More broadly, a crucial next step is to move beyond robust prediction towards robust decision-making, integrating the rich, reliable predictive distributions produced by our framework into decision-theoretic pipelines for high-stakes applications.

References

- Arce, P. G., Naveiro, R., and Insua, D. R. (2025). Evasion attacks against bayesian predictive models. In Chiappa, S. and Magliacane, S., editors, *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 184–202. PMLR.
- Balaji, Y., Goldstein, T., and Hoffman, J. (2019). Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.
- Cai, Q.-Z., Du, M., Liu, C., and Song, D. (2018). Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*.
- Carreau, M., Naveiro, R., and Caballero, W. N. (2025). Poisoning bayesian inference via data deletion and replication. *arXiv preprint arXiv:2503.04480*.
- Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR.
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Wine Quality. UCI Machine Learning Repository.
- Dalvi, N., Domingos, P., Mausam, Sumit, S., and Verma, D. (2004). Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 99–108.
- De Palma, G., Kiani, B., and Lloyd, S. (2021). Adversarial robustness guarantees for random deep neural networks. In *International Conference on Machine Learning*, pages 2522–2534. PMLR.
- Dong, J., Qu, X., Wang, Z. J., and Ong, Y.-S. (2024). Enhancing adversarial robustness via uncertainty-aware distributional adversarial training. *arXiv preprint arXiv:2411.02871*.
- Feng, Y., Rudner, T. G., Tsilivis, N., and Kempe, J. (2024). Attacking bayes: On the adversarial robustness of bayesian neural networks. *arXiv preprint arXiv:2404.19640*.
- Gallego, V., Naveiro, R., and Insua, D. R. (2019). Reinforcement learning under threats. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9939–9940.
- Gallego, V., Naveiro, R., Redondo, A., Ríos Insua, D., and Ruggeri, F. (2024). Protecting classifiers from attacks. *Statistical Science*, 39(3):449–468.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Insua, D. R., Naveiro, R., Gallego, V., and Poulos, J. (2023). Adversarial machine learning: Bayesian perspectives. *Journal of the American Statistical Association*, 118(543):2195–2206.
- Joseph, A., Melson, B., Rubinstein, B., and Tygar, J. (2019). *Adversarial Machine Learning*. Cambridge University Press.
- Kannan, H., Kurakin, A., and Goodfellow, I. (2018). Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.
- LeCun, Y., Cortes, C., and Burges, C. (1998). THE MNIST DATABASE of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- Lin, G., Li, C., Zhang, J., Tanaka, T., and Zhao, Q. (2024). Adversarial training on purification (atop): Advancing both robustness and generalization. *arXiv preprint arXiv:2401.16352*.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.
- Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. (2022). Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*.
- Papamakarios, G., Sterratt, D., and Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd international conference on artificial intelligence and statistics*, pages 837–848. PMLR.
- Tsanas, A. and Xifara, A. (2012). Energy efficiency. UCI Machine Learning Repository.
- Vorobeichyk, Y. and Kantarcioglu, M. (2019). *Adversarial Machine Learning*. Morgan & Claypool.
- Ye, N. and Zhu, Z. (2018). Bayesian adversarial learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6892–6901. Curran Associates Inc.
- Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR.
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z.,

et al. (2023). A survey of large language models.
arXiv preprint arXiv:2303.18223, 1(2).

Supplementary Material for: A unified Bayesian framework for adversarial robustness

A ON THE INCONSISTENCY OF BAYESIAN ADVERSARIAL LEARNING

This section provides a formal analysis of the probabilistic model underlying the Bayesian Adversarial Learning (BAL) framework (Ye and Zhu, 2018). We demonstrate that this framework is based on a set of conditional distributions that are not mathematically consistent with any single, valid joint probability distribution and, thus, does not yield a valid posterior. Our critique is formal in nature and does not comment on the empirical utility of the resulting algorithm. Rather, we show that the framework is not strictly Bayesian as its proposed sampler does not target a valid posterior distribution.

BAL models the problem as a game between a learner and an adversary. Given a clean training set $\mathcal{D}$, the adversary’s goal is to generate a perturbed dataset $\tilde{\mathcal{D}}$ to mislead the learner. The learner, in turn, seeks to perform robust posterior inference on its model parameters θ . To approximate a robust posterior, BAL proposes a Gibbs sampler that alternates between two conditional distributions, each representing one player’s strategy:

- The *Learner’s conditional*, which updates the model parameters θ to *minimize* the loss L on the current perturbed dataset $\tilde{\mathcal{D}}$

$$p(\theta|\tilde{\mathcal{D}}) \propto \exp\{-L(\tilde{\mathcal{D}}; \theta)\} \cdot p(\theta).$$

- The *Adversary’s conditional*, which generates a perturbed dataset $\tilde{\mathcal{D}}$ to *maximize* the same loss L

$$p(\tilde{\mathcal{D}}|\theta, \mathcal{D}) \propto \exp\{+L(\tilde{\mathcal{D}}; \theta) - \alpha \cdot c(\tilde{\mathcal{D}}, \mathcal{D})\}, \quad (8)$$

where $c(\tilde{\mathcal{D}}, \mathcal{D})$ is the perturbation cost, and the hyperparameter α balances this cost against the learner’s loss.

We prove that these two conditionals are inconsistent.

Proposition 1. *The conditional distributions for the learner and the adversary defined in the BAL framework cannot be derived from a single, valid joint probability distribution.*

Proof. Proceed by contradiction. Suppose there exists a single joint distribution $p(\theta, \tilde{\mathcal{D}}|\mathcal{D})$ that is consistent with both conditionals. The BAL paper’s Gibbs sampler implicitly assumes the conditional independence $p(\theta|\tilde{\mathcal{D}}, \mathcal{D}) = p(\theta|\tilde{\mathcal{D}})$.

For our assumed joint distribution to be consistent with the learner’s conditional, it must have the following general form

$$p(\theta, \tilde{\mathcal{D}}|\mathcal{D}) = \exp\{-L(\tilde{\mathcal{D}}; \theta)\} \cdot p(\theta) \cdot g(\tilde{\mathcal{D}}, \mathcal{D}),$$

where $g(\tilde{\mathcal{D}}, \mathcal{D})$ is a function that is independent of θ .

Next, let us derive the adversary’s conditional distribution from this same assumed joint distribution. For a fixed θ , this conditional is proportional to the joint

$$p(\tilde{\mathcal{D}}|\theta, \mathcal{D}) \propto p(\theta, \tilde{\mathcal{D}}|\mathcal{D}) \propto \exp\{-L(\tilde{\mathcal{D}}; \theta)\} \cdot g(\tilde{\mathcal{D}}, \mathcal{D}), \quad (9)$$

where the term $p(\theta)$ is absorbed into the proportionality constant.

We then have two expressions, (8) and (9), for the adversary’s conditional that must be consistent. This requires

$$\exp\{+L(\tilde{\mathcal{D}}; \theta) - \alpha \cdot c(\tilde{\mathcal{D}}, \mathcal{D})\} \propto \exp\{-L(\tilde{\mathcal{D}}; \theta)\} \cdot g(\tilde{\mathcal{D}}, \mathcal{D}).$$

By rearranging the terms, we find what $g(\tilde{\mathcal{D}}, \mathcal{D})$ must be proportional to

$$g(\tilde{\mathcal{D}}, \mathcal{D}) \propto \exp\{2 \cdot L(\tilde{\mathcal{D}}; \theta) - \alpha \cdot c(\tilde{\mathcal{D}}, \mathcal{D})\}.$$

This expression for $g(\tilde{\mathcal{D}}, \mathcal{D})$ depends on the model parameters θ through the loss term $L(\tilde{\mathcal{D}}; \theta)$, contradicting our initial requirement that $g(\tilde{\mathcal{D}}, \mathcal{D})$ must be independent of θ .

Therefore, the initial assumption is false. Consequently, the Gibbs sampler defined by these conditionals does not converge to a valid posterior distribution. $\square$

B DISTINCTNESS OF THE REACTIVE AND PROACTIVE PREDICTIVE DISTRIBUTIONS

Using a counterexample, this section provides a formal proof that the posterior predictive distributions derived from the proactive (defense during training) and reactive (defense during operations) strategies are, in general, different. We construct a simple linear-Gaussian model where most relevant quantities can be computed analytically, revealing differences between both approaches.

Model Setup. Define a simple model with the following components: a clean univariate covariate model, $x \sim \mathcal{N}(0, \sigma_x^2)$ with known variance σ_x^2 (thus $\phi = \sigma_x^2$); a known linear corruption channel that is independent from model parameters, $x' = (1 + \varepsilon)x + \nu$ where $\nu \sim \mathcal{N}(0, \sigma_\delta^2)$ with known σ_δ^2 ; a linear model, $y \mid x, \theta \sim \mathcal{N}(\theta x, \sigma_y^2)$ with known σ_y^2 ; and a Gaussian prior on the single unknown parameter, $\theta \sim \mathcal{N}(0, \sigma_\theta^2)$.

Strategy 1: Reactive Defense. Suppose that the model is trained on a single clean data pair $\mathcal{D} = \{(x_1, y_1)\}$. Upon observing the possibly corrupted covariate x' , the robust posterior predictive distribution (PPD) in equation (1) from the main paper can be written as

$$p(y \mid x', \mathcal{D}) = \iint p(y \mid x, \theta) p(x \mid x') p(\theta \mid \mathcal{D}) dx d\theta.$$

In this setup, the posterior on the parameter is Gaussian, $p(\theta \mid \mathcal{D}) = \mathcal{N}(\theta; \mu_{\text{RE}}, v_{\text{RE}})$, with

$$\mu_{\text{RE}} = \frac{\sigma_\theta^2 x_1 y_1}{\sigma_y^2 + \sigma_\theta^2 x_1^2}, \quad v_{\text{RE}} = \frac{\sigma_\theta^2 \sigma_y^2}{\sigma_y^2 + \sigma_\theta^2 x_1^2}.$$

The posterior $p(x \mid x')$ for the latent clean covariate is also Gaussian, with mean $\mathbb{E}[x \mid x'] = \lambda x'$ and variance $\text{Var}(x \mid x') = \tau^2$, where

$$\lambda = \frac{(1 + \varepsilon)\sigma_x^2}{(1 + \varepsilon)^2\sigma_x^2 + \sigma_\delta^2}, \quad \tau^2 = (1 - \lambda(1 + \varepsilon))\sigma_x^2.$$

The robust PPD is non-Gaussian. Its first two moments are

$$\begin{aligned} \mathbb{E}_{\text{RE}}[y \mid x', \mathcal{D}] &= \lambda x' \mu_{\text{RE}}, \\ \text{Var}_{\text{RE}}[y \mid x', \mathcal{D}] &= \sigma_y^2 + \lambda^2 x'^2 v_{\text{RE}} + \tau^2 (\mu_{\text{RE}}^2 + v_{\text{RE}}). \end{aligned}$$

Strategy 2: Proactive Defense. This strategy embeds the corruption channel within the training likelihood. Following equation (3) from the main paper, the resulting “corruption-aware” likelihood is $p(y_1 \mid x_1, \theta) = \mathcal{N}(y_1; (1 + \varepsilon)\theta x_1, \sigma_y^2 + \theta^2 \sigma_\delta^2)$. As a consequence, the posterior $p(\theta \mid \mathcal{D})$ is non-Gaussian. At test time, the PPD is obtained by integrating over this posterior

$$p(y \mid x', \mathcal{D}) = \int \mathcal{N}(y; \theta x', \sigma_y^2) p(\theta \mid \mathcal{D}) d\theta.$$

Denoting the moments of the non-Gaussian posterior as μ_{PR} and v_{PR} , the predictive moments are

$$\begin{aligned} \mathbb{E}_{\text{PR}}[y \mid x', \mathcal{D}] &= x' \mu_{\text{PR}}, \\ \text{Var}_{\text{PR}}[y \mid x', \mathcal{D}] &= \sigma_y^2 + x'^2 v_{\text{PR}}. \end{aligned}$$

Conclusion. These distributions are clearly different. Indeed, assume that both PPDs are identical. Then, from the condition of equal variances, we must have

$$\sigma_y^2 + \lambda^2 x'^2 v_{\text{RE}} + \tau^2 (\mu_{\text{RE}}^2 + v_{\text{RE}}) = \sigma_y^2 + x'^2 v_{\text{PR}}.$$

Rearranging this expression yields

$$x'^2 (v_{\text{PR}} - \lambda^2 v_{\text{RE}}) = \tau^2 (\mu_{\text{RE}}^2 + v_{\text{RE}}).$$

For this equality to hold for all possible test inputs x' , the left-hand side, which is a function of x' , must equal the right-hand side, which is a constant. This is only possible if both sides are zero. For the right-hand side to be zero, given that $\mu_{\text{RE}}^2 + v_{\text{RE}} > 0$, it is necessary that $\tau^2 = 0$. This condition corresponds to the degenerate case of a noise-free, perfectly invertible corruption channel. In any non-degenerate case where channel uncertainty exists ($\tau^2 > 0$), the equality leads to a contradiction. Therefore, both predictive distributions are demonstrably distinct.

C FORMAL CONNECTION WITH RANDOMIZED SMOOTHING

This section derives a connection between our protection during operations approach and *randomized smoothing* (Cohen et al., 2019). This is a certification technique that provides high-probability robustness guarantees for a base classifier. It works by constructing a new, smoothed classifier, g , whose prediction at a point $\mathbf{x}'$ is the class most likely to be returned by a base classifier, f , when its input is perturbed with isotropic Gaussian noise. Formally, the smoothed classifier prediction is

$$g(\mathbf{x}') = \arg \max_c \mathbb{P}_{\delta \sim \mathcal{N}(0, \sigma^2 I)} (f(\mathbf{x}' + \delta) = c).$$

Proposition 2. *The Bayesian predictive distribution of our protection during operations is equivalent to the predictive probability of a randomized smoothed classifier under the following modeling choices:*

- *Maximum A Posteriori (MAP) estimates are utilized for both θ and ϕ .*
- *The prior over the latent input is uniform, i.e., $p(\mathbf{x}|\phi) \propto \text{constant}$.*
- *The adversarial channel is isotropic Gaussian, $p(\mathbf{x}'|\mathbf{x}, \theta) = \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 I)$.*

Proof. Let us begin with the full Bayesian predictive distribution

$$p(y|\mathbf{x}', \mathcal{D}) = \mathbb{E}_{(\theta, \phi) | \mathcal{D}} [\mathbb{E}_{\mathbf{x}|\mathbf{x}', \theta, \phi} [p(y|\mathbf{x}, \theta)]] .$$

First, we apply the MAP approximation. This replaces the outer expectation over the parameters posterior with a single evaluation at the MAP estimates ($\theta_{\text{MAP}}, \phi_{\text{MAP}}$)

$$p(y|\mathbf{x}', \mathcal{D}) \approx p(y|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}}) = \mathbb{E}_{\mathbf{x}|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}}} [p(y|\mathbf{x}, \theta_{\text{MAP}})] \quad (10)$$

Next, derive the distribution for the remaining expectation, $p(\mathbf{x}|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}})$. By Bayes' theorem

$$p(\mathbf{x}|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}}) \propto p(\mathbf{x}'|\mathbf{x}, \theta_{\text{MAP}}) p(\mathbf{x}|\phi_{\text{MAP}}).$$

Substituting the specified models,

$$p(\mathbf{x}|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}}) \propto \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 I) \cdot \text{constant}.$$

The Gaussian density is symmetric in its arguments, meaning that the kernel $\exp(-\frac{1}{2\sigma^2} \|\mathbf{x}' - \mathbf{x}\|^2)$ is proportional to the kernel of a Gaussian distribution for $\mathbf{x}$ centered at $\mathbf{x}'$. Since $p(\mathbf{x}|\mathbf{x}', \theta, \phi)$ must be a normalized probability distribution, it is necessarily the Gaussian

$$p(\mathbf{x}|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}}) = \mathcal{N}(\mathbf{x}; \mathbf{x}', \sigma^2 I).$$

Therefore, the latent input $\mathbf{x}$ is a noisy version of the input, $\mathbf{x} \sim \mathbf{x}' + \mathcal{N}(0, \sigma^2 I)$. Finally, substituting this result back into equation (10), we obtain

$$p(y|\mathbf{x}', \theta_{\text{MAP}}, \phi_{\text{MAP}}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{N}(\mathbf{x}', \sigma^2 I)} [p(y|\mathbf{x}, \theta_{\text{MAP}})] .$$

This expression is the definition of the predictive probability for a randomized smoothed classifier with base classifier $f(\cdot) = p(\cdot | \cdot, \theta_{\text{MAP}})$ and smoothing noise $\mathcal{N}(0, \sigma^2 I)$. $\square$

D DERIVATION OF THE ONLINE ADAPTIVE DEFENSE

In the main paper, we derived a defense based on the simplifying assumption that the posterior over the model parameters is fixed at test time, i.e., $p(\theta, \phi | \mathbf{x}'_j, \mathcal{D}) \approx p(\theta, \phi | \mathcal{D})$. This section derives an alternative strategy by avoiding this assumption. This results in a more adaptive defense that uses the new, corrupted data $\mathbf{x}'_j$ to update its belief about the model parameters θ and ϕ .

D.1 Derivation of the Predictive Distribution

Begin by writing the full PPD for a new data point y_j (equation (1) in the paper)

$$p(y_j | \mathbf{x}'_j, \mathcal{D}) = \iiint p(y_j | \mathbf{x}_j, \theta) p(\mathbf{x}_j, \theta, \phi | \mathbf{x}'_j, \mathcal{D}) d\mathbf{x}_j d\theta d\phi.$$

Using the chain rule and Bayes' theorem, this integral can be expanded. After canceling the $p(\mathbf{x}'_j | \theta, \phi)$ terms, the expression simplifies to

$$p(y_j | \mathbf{x}'_j, \mathcal{D}) = \frac{\mathbb{E}_{\theta, \phi | \mathcal{D}} [\mathbb{E}_{\mathbf{x}_j | \phi} [p(y_j | \mathbf{x}_j, \theta) p(\mathbf{x}'_j | \mathbf{x}_j, \theta)]]}{\mathbb{E}_{\theta, \phi | \mathcal{D}} [\mathbb{E}_{\mathbf{x}_j | \phi} [p(\mathbf{x}'_j | \mathbf{x}_j, \theta)]]}.$$

This final form is a ratio of expectations, which we approximate using Monte Carlo.

D.2 Approximation with the Empirical Distribution

To make this computation tractable, we make two approximations:

1. We approximate the posterior $p(\theta, \phi | \mathcal{D})$ with a set $\{(\theta_s, \phi_s)\}_{s=1}^S$ of S samples from it.
2. We replace the generative model $p(\mathbf{x}_j | \phi)$ with the empirical distribution of the N training data points, $\{\mathbf{x}_i\}_{i=1}^N$.

Under these approximations, the integrals in the PPD become finite sums so that the PPD is approximated as the ratio of these sums

$$p(y_j | \mathbf{x}'_j) \approx \frac{\sum_{s=1}^S \sum_{i=1}^N p(y_j | \mathbf{x}_i, \theta_s) p(\mathbf{x}'_j | \mathbf{x}_i, \theta_s)}{\sum_{s'=1}^S \sum_{k=1}^N p(\mathbf{x}'_j | \mathbf{x}_k, \theta_{s'})}$$

This is a single, globally normalized weighted sum over all $S \times N$ hypotheses. While this ‘‘plug-in’’ ratio estimator is biased for finite samples, it is consistent, converging to the true PPD as the number of samples S and data points N approaches infinity.

This ‘‘Online’’ defense is more adaptive than the reactive defense in the paper. The mathematical proof of this lies in how it implicitly favors certain parameter samples θ_s drawing on the new evidence $\mathbf{x}'_j$. To wit, define the prediction from a single parameter sample θ_s as

$$p(y_j | \mathbf{x}'_j, \theta_s) = \frac{\sum_{i=1}^N p(y_j | \mathbf{x}_i, \theta_s) p(\mathbf{x}'_j | \mathbf{x}_i, \theta_s)}{\sum_{k=1}^N p(\mathbf{x}'_j | \mathbf{x}_k, \theta_s)},$$

and its corresponding marginal likelihood for the new data $\mathbf{x}'_j$ as $L_s(\mathbf{x}'_j) = \sum_{i=1}^N p(\mathbf{x}'_j | \mathbf{x}_i, \theta_s)$. We rewrite the Online PPD as

$$p_{\text{ON}}(y_j | \mathbf{x}'_j) \approx \sum_{s=1}^S \left(\frac{L_s(\mathbf{x}'_j)}{\sum_{s'=1}^S L_{s'}(\mathbf{x}'_j)} \right) \cdot p(y_j | \mathbf{x}'_j, \theta_s),$$

while, the probability corresponding to the reactive defense in the paper is

$$p_{\text{RE}}(y_j | \mathbf{x}'_j) \approx \sum_{s=1}^S \frac{1}{S} \cdot p(y_j | \mathbf{x}'_j, \theta_s).$$

This shows that the Online PPD is a weighted average of the component predictions with weights proportional to their marginal likelihood, while in the reactive defense weights were constant. Thus, this is more adaptive: it uses the new evidence $\mathbf{x}'_j$ to re-weight each posterior sample θ_s , giving more influence to the parameters that provide a better explanation for observed $\mathbf{x}'_j$.

E FORMAL CONNECTION WITH ADVERSARIAL TRAINING

This section derives a formal connection between our proactive defense and adversarial training.

Proposition 3. *The standard minimax formulation of Adversarial Training (AT) is a deterministic, point-estimate limit of the proposed proactive Bayesian defense.*

Proof. We start with the log-robust-posterior for the parameters θ from Section 3.3 in the main text

$$\log p(\theta|\mathcal{D}) \propto \log p(\theta) + \sum_{i=1}^N \log \mathbb{E}_{\mathbf{x}'_i \sim p(\cdot|\mathbf{x}_i, \theta)} [p(y_i|\mathbf{x}'_i, \theta)]. \quad (11)$$

Consider the limit where the stochastic channel collapses to a deterministic function that outputs the single worst-case attack $\mathbf{x}_i^*$. Define the loss $\mathcal{L}(f_\theta(\mathbf{x}), y)$ as the negative log-likelihood, $-\log p(y|\mathbf{x}, \theta)$. The worst-case attack is then

$$\mathbf{x}_i^* = \arg \max_{\mathbf{z} \in \mathcal{B}(\mathbf{x}_i, \epsilon)} \mathcal{L}(f_\theta(\mathbf{z}), y_i).$$

Setting the channel to be a Dirac delta function, $p(\mathbf{x}'_i|\mathbf{x}_i, \theta) = \delta(\mathbf{x}'_i - \mathbf{x}_i^*)$, collapses the expectation in (11), simplifying the log-posterior to

$$\log p(\theta|\mathcal{D}) \propto \log p(\theta) + \sum_{i=1}^N \log p(y_i|\mathbf{x}_i^*, \theta). \quad (12)$$

Next, we replace the fully Bayesian objective of finding the entire posterior with that of finding a point estimate via Maximum a Posteriori (MAP) estimation. The MAP estimate θ_{MAP} is the mode of the posterior (12)

$$\theta_{\text{MAP}} = \arg \max_{\theta} \left(\log p(\theta) + \sum_{i=1}^N \log p(y_i|\mathbf{x}_i^*, \theta) \right).$$

Using our definition of the loss function, maximizing the log-posterior is equivalent to minimizing the negative log-posterior, which yields a regularized loss minimization problem

$$\theta_{\text{MAP}} = \arg \min_{\theta} \left(\sum_{i=1}^N \mathcal{L}(f_\theta(\mathbf{x}_i^*), y_i) - \log p(\theta) \right).$$

$$\theta_{\text{MAP}} = \arg \min_{\theta} \sum_{i=1}^N \mathcal{L}(f_\theta(\mathbf{x}_i^*), y_i).$$

Finally, substituting the definition of $\mathbf{x}_i^*$ yields the standard minimax AT formulation

$$\theta_{\text{AT}} = \arg \min_{\theta} \sum_{i=1}^N \max_{\mathbf{z} \in \mathcal{B}(\mathbf{x}_i, \epsilon)} \mathcal{L}(f_\theta(\mathbf{z}), y_i).$$

Thus, AT is recovered as a deterministic, point-estimate limit of our proactive defense. $\square$

Notably, if instead of an improper uniform prior for θ , one chooses a prior of the form $(p(\theta|\mathcal{D}) \propto \exp(-\lambda \sum_i |f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}'_i)|^2))$, the resulting MAP objective recovers the Adversarial Logit Pairing regularizer (Kannan et al., 2018).

F ALTERNATIVE GRADIENT ESTIMATION FOR THE PROACTIVE DEFENSE

As the main text describes, the proactive defense maximizes the ELBO in Eq. (7) of Section 3.3

$$\mathcal{L}(\psi) = \mathbb{E}_{\theta \sim q_\psi(\theta)} \left[\sum_{i=1}^N \log \mathbb{E}_{\mathbf{x}'_i \sim p(\cdot|\mathbf{x}_i, \theta)} [p(y_i|\mathbf{x}'_i, \theta)] + \log p(\theta) - \log q_\psi(\theta) \right].$$

With a reparameterizable posterior $\theta = f(\psi, \epsilon)$, $\epsilon \sim p(\epsilon)$, the gradient becomes

$$\nabla_\psi \mathcal{L}(\psi) = \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[\sum_{i=1}^N \nabla_\psi \log \mathbb{E}_{\mathbf{x}'_i \sim p(\cdot|\mathbf{x}_i, \theta)} [p(y_i|\mathbf{x}'_i, f(\psi, \epsilon))] \right] - \nabla_\psi \text{KL}(q_\psi(\theta) \| p(\theta)). \quad (13)$$

The log term is problematic because, even with the reparameterization, its pathwise gradient is a ratio of expectations; plugging in sample averages gives a biased ratio-of-means. If we further assume the adversarial channel is also reparameterizable, $\mathbf{x}' = h(\mathbf{x}, \theta, \boldsymbol{\eta})$, $\boldsymbol{\eta} \sim p(\boldsymbol{\eta})$, then

$$\nabla_\psi \log \mathbb{E}_{\mathbf{x}'_i \sim p(\cdot|\mathbf{x}_i, \theta)} p(y_i|\mathbf{x}'_i, f(\psi, \epsilon)) = \frac{\mathbb{E}_{\boldsymbol{\eta} \sim p(\boldsymbol{\eta})} [\nabla_\psi p(y_i|h(\mathbf{x}_i, \theta, \boldsymbol{\eta}), f(\psi, \epsilon))]}{\mathbb{E}_{\boldsymbol{\eta} \sim p(\boldsymbol{\eta})} [p(y_i|h(\mathbf{x}_i, \theta, \boldsymbol{\eta}), f(\psi, \epsilon))]}.$$

While this ratio could be estimated unbiasedly using methods like Multilevel Monte Carlo, we propose a biased but consistent estimator built directly from samples: for each gradient step we (i) draw S noise samples $\epsilon_s \sim p(\epsilon)$, (ii) compute the corresponding parameters $\theta_s = f(\psi, \epsilon_s)$, (iii) for each data point i draw K adversarial noises $\boldsymbol{\eta}_{isk} \sim p(\boldsymbol{\eta})$ and construct $\mathbf{x}'_{isk} = h(\mathbf{x}_i, \theta_s, \boldsymbol{\eta}_{isk})$, (iv) evaluate $p_{isk} = p(y_i|\mathbf{x}'_{isk}, \theta_s)$ and the pathwise gradients $\nabla_\psi \log p_{isk}$, and (v) plug these into the ratio. In practice, this yields

$$\hat{g}_{i,s}(\psi) = \sum_{k=1}^K \tilde{w}_{isk} \nabla_\psi \log p_{isk}, \quad \tilde{w}_{isk} = \frac{p_{isk}}{\sum_{k'=1}^K p_{isk'}}.$$

Finally, the approximate gradient of the ELBO is

$$\nabla_\psi \mathcal{L}(\psi) \approx \frac{N}{|\mathcal{B}|S} \sum_{s=1}^S \sum_{i \in \mathcal{B}} \hat{g}_{i,s}(\psi) - \frac{1}{S} \sum_{s=1}^S \nabla_\psi [\log q_\psi(\theta_s) - \log p(\theta_s)],$$

where $\mathcal{B}$ is a mini-batch. This estimator explicitly samples $(\epsilon_s, \theta_s, \boldsymbol{\eta}_{isk})$ and substitutes them into the ratio, producing a biased but consistent approximation whose bias vanishes as $K, S \rightarrow \infty$,

$$\hat{g}_{i,s}(\psi) = \sum_{k=1}^K \tilde{w}_{isk} \nabla_\psi \log p_{isk}, \quad \tilde{w}_{isk} = \frac{p_{isk}}{\sum_{k'=1}^K p_{isk'}}.$$

Then, the full gradient estimator is

$$\nabla_\psi \mathcal{L}(\psi) \approx \frac{N}{|\mathcal{B}|S} \sum_{s=1}^S \sum_{i \in \mathcal{B}} \hat{g}_{i,s}(\psi) - \frac{1}{S} \sum_{s=1}^S \nabla_\psi [\log q_\psi(\theta_s) - \log p(\theta_s)].$$

G ADDITIONAL INFORMATION ON EXPERIMENTS

G.1 Experimental Setup

This subsection details the experimental setup used throughout our experiments. Full hyperparameter specifications, including learning rates, batch sizes, decay rates, and number of training iterations, are available in the repository.

G.1.1 Model Architectures

We transform the deterministic architectures described below into Bayesian neural networks by placing prior distributions over all network weights and biases. Specifically, we employ a factorized Gaussian prior with zero mean and unit variance: $p(\theta) = \mathcal{N}(0, I)$.

Inference is performed via variational inference, optimizing a mean-field variational posterior distribution $q_\psi(\theta)$ to approximate the true posterior $p(\theta|\mathcal{D})$, using the Adam optimizer with exponential learning rate decay.

Classification Model For MNIST classification, we employ a convolutional neural network comprising two convolutional blocks followed by a fully connected output layer. Each block consists of a convolutional layer (16 and 32 filters, respectively, and 3×3 kernels), ReLU activation, and 2×2 average pooling. The resulting feature maps are flattened and passed through a dense layer producing logits for the 10 digit classes.

Regression Model For regression tasks, we employ a fully connected feedforward architecture with two hidden layers of 16 neurons each with ReLU activations. The input is processed through these hidden layers before a final dense layer outputs a scalar prediction.

G.1.2 Adversarial Channels

The central modeling choice in our framework is the design of the adversarial channel, which formalizes our assumptions about the adversary. It is crucial to distinguish these channels, which are an integral component of our defense methodology; from the separate attacks used for the final evaluation of a model’s robustness. Each channel defines the stochastic mapping:

$$p(\mathbf{x}' | \mathbf{x}, \theta)$$

that transforms clean inputs $\mathbf{x}$ into adversarial counterparts $\mathbf{x}'$ conditioned on model parameters θ . These channels differ in the degree of adaptivity and stochasticity introduced during training and inference. We now detail the specific channels implemented in our experiments.

Identity Adversary. The identity adversary serves as a baseline, introducing no adversarial bias. It generates simple Gaussian augmentations of the input data

$$\mathbf{x}' = \mathbf{x} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I),$$

where σ is the augmentation standard deviation. This channel is primarily used to assess the baseline stability of the model under random perturbations without adversarial intent.

Naive One-Step Adversary. The one-step adversary implements a fast gradient-based perturbation strategy akin to one iteration of PGD. It computes adversarial examples by taking a single gradient step on the loss surface with respect to the input,

$$\mathbf{x}' = \mathbf{x} + \epsilon \frac{\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \theta)}{\|\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \theta)\|_2} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I),$$

where $\mathcal{L}$ is the task loss (e.g. cross-entropy, CW, Entropy, RMSE). These adversarial examples are then further augmented with Gaussian noise to increase variability.

Mixed One-Step Adversary. The mixed one-step adversary combines both clean and adversarial augmentations within each batch. Half of the augmented samples are generated using the one-step perturbations described above, while the remaining half are drawn from the identity adversary. This hybrid scheme introduces a controlled mixture of adversarial and non-adversarial variability, enabling smoother training dynamics and improved generalization under partially adversarial conditions.

PGD Adversary. The projected gradient descent adversary implements a stronger, iterative gradient-based attack: starting from either the clean input or a small random perturbation, it performs T iterative updates

$$\begin{aligned}\mathbf{x}^{t+1} &\leftarrow \Pi_{B_\epsilon}(\mathbf{x}^t + \alpha \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^t, \theta)), \\ \mathbf{x}^T &\leftarrow \mathbf{x}^T + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I),\end{aligned}$$

where α is the step size, Π_{B_ϵ} denotes projection onto the allowed perturbation ball. PGD attacks typically include a random initialization within B_ϵ to avoid gradient masking and are substantially more computationally expensive than one-step methods, but they produce stronger adversarial examples for both training and evaluation.

General Adversarial Channel. During training or evaluation, a general adversarial mechanism probabilistically selects one of the available channels according to predefined mixture weights. Formally, given a set of adversarial specifications (A_k, π_k) , where (A_k) denotes an adversary and π_k its selection probability, the active adversarial channel is drawn as

$$A \sim \text{Categorical}(\pi_1, \dots, \pi_K),$$

and used to generate perturbed inputs $(\mathbf{x}' = A(\mathbf{x}, y, \theta))$. This formulation enables stochastic sampling over multiple adversarial mechanisms, supporting both fixed and learned mixtures of perturbation strategies.

The MIX adversary is implemented as an instance of the general adversarial channel: it is a mixture of an identity (Gaussian-augmentation) component, sampled with probability 0.4, and, for computational ease, a one-step adversary with three different loss objectives, cross-entropy, CW, and an entropy-based loss, each sampled with probability 0.2 using a fixed $\epsilon = 1$. This combination yields a diverse set of perturbations.

G.1.3 Learned Adversarial Perturbation Models

As one instantiation of the adversarial channel detailed in Section 3, we develop learned adversarial models that generate perturbations conditioned on both input and label. These generative models parameterize perturbation distributions rather than producing deterministic ones, enabling gradient-based adversarial training against stochastic attacks. Training follows a GAN-style procedure. At each batch iteration, the adversarial model is first updated to maximize a loss function associated with model performance (cross-entropy for classification and RMSE for regression). Subsequently, the main model is adversarially trained using samples drawn from $p(\mathbf{x}'|\mathbf{x}, \theta)$ generated by the adversarial model.

Convolutional Adversarial Model For MNIST, we design a convolutional perturbation generator preserving spatial structure. Given input image $\mathbf{x}$ and label y (one-hot encoded), the image is processed through two convolutional layers (16 filters, 3×3 kernels, same padding) with ReLU activations, followed by a third convolutional layer matching the input channel dimension. The spatial features are flattened, combined with a learned label projection via element-wise addition, and reshaped to the original dimensions. This produces a mean perturbation $\boldsymbol{\mu}$ added to the input ($\boldsymbol{\mu} + \mathbf{x}$) and per-pixel log-standard deviations $\log \boldsymbol{\sigma}$ from an additional convolutional layer, jointly parameterizing a Gaussian perturbation distribution.

Fully Connected Adversarial Model For regression datasets, we employ a fully connected perturbation generator that concatenates a learned projection of the scalar label y with input features $\mathbf{x}$. This representation is processed through two hidden layers (128 units each) with ReLU activations, followed by a final layer matching the input dimensionality. Analogously to the convolutional variant, this produces a mean perturbation $\boldsymbol{\mu}$ (added to $\mathbf{x}$) and per-feature log-standard deviations $\log \boldsymbol{\sigma}$ through separate projections, parameterizing a Gaussian perturbation distribution for diverse, label-conditioned adversarial perturbations.

G.2 Computational Overhead

As the robustification methodologies present distinct trade-offs between training and inference efficiency, let us evaluate the computational costs associated with each robustification approach.

The proactive defense modifies the training objective, resulting in moderately increased training time with no test-time overhead beyond standard Bayesian inference. Conversely, the reactive defense preserves standard

Table 5: Training and inference times (in ms) across datasets. Training times are reported per batch iteration, and inference times per sample, both averaged over 100 iterations. Robust Training corresponds to the MIX model (proactive defense), while Robust Inference denotes the onPure model (online reactive defense), the two defenses with the highest computational cost.

Dataset	Standard Training	Robust Training	Standard Inference	Robust Inference
MNIST	2.48	4.69	1.60	2470
Wine Quality	0.75	0.70	0.63	3.39
Energy Efficiency	0.74	0.66	0.65	3.47
California Housing	0.74	0.68	0.62	3.44

training costs but introduces significant test-time computation and memory overhead. The inference overhead depends critically on the number of parameter (S) and input (N) samples used in the weighted estimate. To reduce memory cost, the estimate is computed on a random subsample of the training set rather than on the full dataset. With our default configuration ($S = 5$, $N = 100$), the reactive defense incurs substantial overhead for high-dimensional inputs (over $1000\times$ for MNIST images) but remains competitive for low-dimensional regression tasks (approximately $5\times$ overhead). Additionally, the reactive defense requires maintaining the training set in memory simultaneously, leading to considerable memory requirements that scale with input dimensionality.

Table 5 summarizes training and inference times across all datasets and methodologies. Note that standard training has not been as extensively optimized as the robust training procedures, which may explain the comparable or even superior throughput observed for robust training on regression datasets. These results enable practitioners to select the appropriate approach based on their computational constraints and data characteristics: the proactive defense is preferable when low-latency inference is critical or when working with high-dimensional inputs, while the reactive defense is viable for low-dimensional problems where its modest inference overhead and memory requirements may be acceptable given the simplified training procedure.

All experiments reported in this section were conducted on a single NVIDIA A100 with 82GB memory.

G.3 Additional Results

Figures 6 and 7 present performance metrics against attack intensity for PGD1 and ENT attacks, respectively. Figure 6 shows results consistent with those reported in Figure 3 of the main text, confirming the same qualitative trends across all defenses. The same qualitative trends remain in Figure 7, with the main difference being that AT performs noticeably worse than the undefended BL, showing lower accuracy and higher NLL values across all perturbation strengths. This suggests that the entropy-targeted perturbations exploit weaknesses in the AT model’s predictive calibration, leading to poorer overall performance despite its nominal robustness under standard attacks.

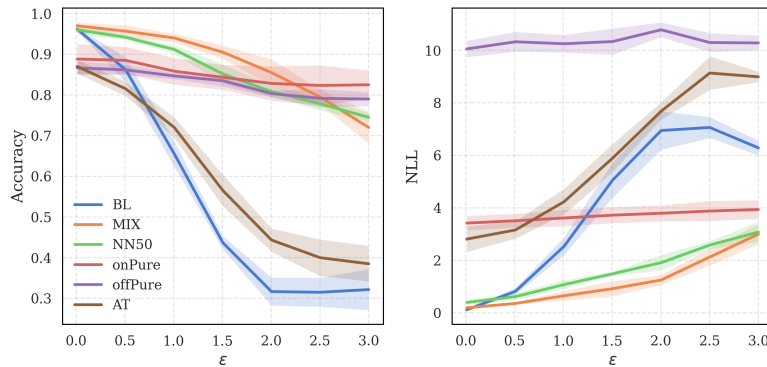


Figure 6: Accuracy and NLL against PGD1 attack.

Figures 8 and 9 show illustrative examples of adversarial attacks and the corresponding predictive distributions for identical inputs from the MNIST dataset. The BL model is easily misled by most perturbations, often producing incorrect predictions or displaying high uncertainty even when the correct class remains among the

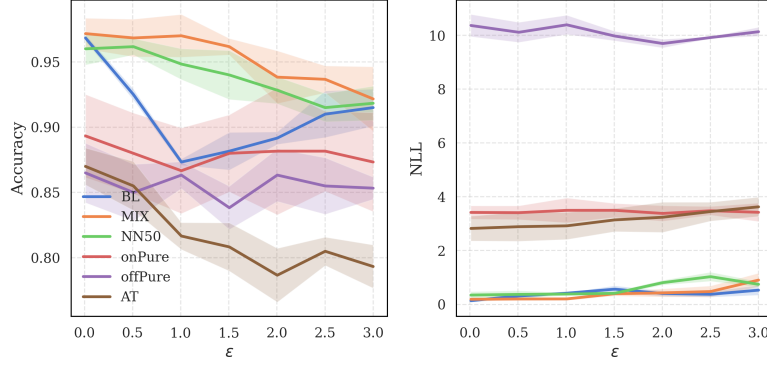


Figure 7: Accuracy and NLL against ENT attack.

top outputs. In contrast, the MIX model shows clear resilience to these attacks, maintaining reliable predictions. Notably, after being attacked, the BL model’s probability mass becomes more dispersed across classes, reflecting degraded calibration and reduced reliability in the correct label, whereas the MIX model preserves a sharper and more consistent predictive distribution.

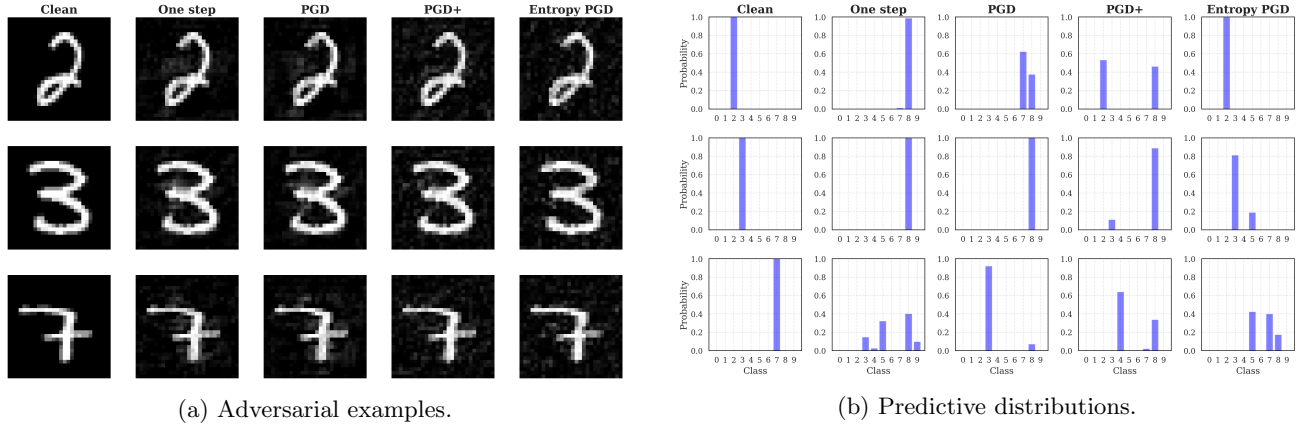
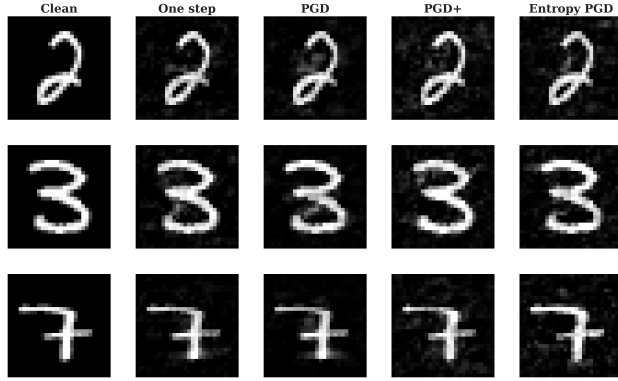


Figure 8: Attacks to BL model.

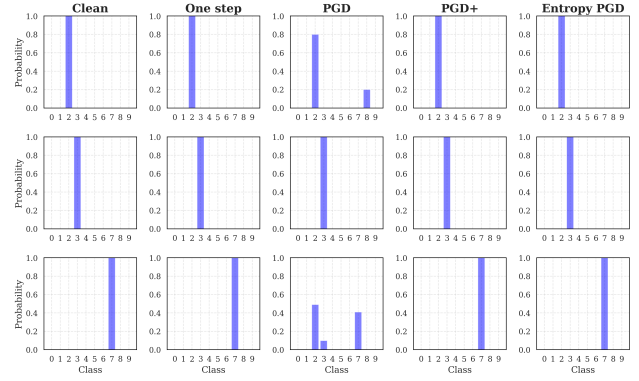
Table 6 presents additional regression results complementing those in the main text. Overall, the same qualitative patterns hold. The BL model performs well on clean data but degrades sharply under attack, while AT exhibits unstable behavior, performing worse than BL in most adversarial settings. NN50 maintains the most consistent performance across attacks, achieving the lowest NLLs and stable RMSE values. MIX remains competitive, showing slightly higher NLLs but comparable robustness. As before, purification-based methods display clear trade-offs: onPure achieves reasonable RMSEs yet suffers from moderate calibration issues, whereas offPure remains poorly calibrated despite stable errors.

Table 6: RMSEs and NLLs on California dataset at $\epsilon = 2$ under different attack types.

Model	RMSE					NLL				
	Clean	PGD1	PGD	PGD ⁺	ENT	Clean	PGD1	PGD	PGD ⁺	ENT
BL	0.20 (0.01)	0.44 (0.02)	0.43 (0.02)	0.76 (0.02)	0.45 (0.01)	-0.15 (0.08)	2.03 (0.10)	1.96 (0.12)	4.96 (0.29)	1.84 (0.21)
MIX	0.23 (0.01)	0.53 (0.01)	0.52 (0.02)	0.63 (0.01)	0.52 (0.01)	-0.00 (0.03)	1.71 (0.19)	1.73 (0.07)	2.36 (0.16)	1.56 (0.14)
NN50	0.33 (0.01)	0.42 (0.01)	0.41 (0.01)	0.46 (0.02)	0.42 (0.02)	0.33 (0.03)	0.58 (0.05)	0.56 (0.04)	0.70 (0.04)	0.62 (0.05)
onPure	0.44 (0.02)	0.42 (0.01)	0.44 (0.01)	0.45 (0.02)	0.43 (0.02)	2.11 (0.31)	2.23 (0.35)	2.34 (0.43)	2.75 (0.59)	2.41 (0.53)
offPure	0.43 (0.02)	0.43 (0.01)	0.45 (0.02)	0.45 (0.03)	0.45 (0.03)	9.73 (1.06)	10.24 (1.11)	10.20 (0.97)	11.90 (1.16)	10.39 (1.31)
AT	0.31 (0.01)	1.58 (0.03)	1.69 (0.01)	3.82 (0.05)	1.97 (0.04)	0.44 (0.01)	3.49 (0.30)	3.67 (0.27)	14.01 (0.62)	2.91 (0.09)



(a) Adversarial examples.



(b) Predictive distributions.

Figure 9: Attacks to MIX model.