Adversarial robustness of amortized Bayesian inference

Manuel Gloeckler [1] **Michael Deistler** [1] **Jakob H. Macke** [1 2]

Abstract

Bayesian inference usually requires running potentially costly inference procedures separately for every new observation. In contrast, the idea of *amortized* Bayesian inference is to initially invest computational cost in training an inference network on simulated data, which can subsequently be used to rapidly perform inference (i.e., to return estimates of posterior distributions) for new observations. This approach has been applied to many real-world models in the sciences and engineering, but it is unclear how robust the approach is to adversarial perturbations in the observed data. Here, we study the adversarial robustness of amortized Bayesian inference, focusing on simulation-based estimation of multi-dimensional posterior distributions. We show that almost unrecognizable, targeted perturbations of the observations can lead to drastic changes in the predicted posterior and highly unrealistic posterior predictive samples, across several benchmark tasks and a real-world example from neuroscience. We propose a computationally efficient regularization scheme based on penalizing the Fisher information of the conditional density estimator, and show how it improves the adversarial robustness of amortized Bayesian inference.

1. Introduction

Bayesian inference is a commonly used approach for identifying model parameters that are compatible with empirical observations and prior knowledge. Classical Bayesian inference methods such as Markov-chain Monte Carlo (MCMC) can be computationally expensive at test-time, as they rely on repeated evaluations of the likelihood function and, therefore, require a new set of likelihood evaluations for each observation. In contrast, the idea of *amortized* Bayesian inference is to approximate the mapping from observation to posterior distribution by a conditional density estimator, often parameterized as a neural network. Once this density estimation network has been trained, inference on a particular observation can be performed very efficiently, requiring only a single forward-pass through the network. This *amortization* can be achieved by training conditional density estimators on simulated data and framing Bayesian inference as a prediction problem: For *any* observation, the neural network is trained to predict either the posterior directly (Papamakarios & Murray, 2016; Greenberg et al., 2019; Gonçalves et al., 2020; Radev et al., 2020) or a quantity that allows to infer the posterior without further simulations (Papamakarios et al., 2019; Hermans et al., 2020). This approach has several advantages over MCMC methods: It can be used to perform 'simulation-based inference', i.e., applied to models which are only implicitly given as simulators (models which allow to sample the likelihood but not to evaluate it), it does not require the model to be differentiable (as compared to, e.g., Hamiltonian Monte Carlo), and it allows application in high-throughput scenarios (Dax et al., 2021; von Krause et al., 2022; Boelts et al., 2022; Arnst et al., 2022).

However, these benefits come at a cost: the posterior predicted by the neural network will not be exact (Lueckmann et al., 2021), can be overconfident (Hermans et al., 2022), and can be sensitive to misspecified models (Cannon et al., 2022; Schmitt et al., 2022). Here, we study another possible limitation of neural network-based amortized Bayesian inference: It is well known that neural networks can be susceptible to adversarial attacks, i.e., tiny but targeted perturbations to the inputs can lead to vastly different outputs (Szegedy et al., 2014). For amortized Bayesian inference, this would indicate that even minor perturbations in the observed data could lead to entirely different posterior estimates.

Adversarial attacks have become a common technique to evaluate the robustness of ML algorithms. Attacks can be used to assess performance in the presence of small worst-case perturbations, offering valuable insights into how models perform when faced with model misspecification. Furthermore, amortized inference is increasingly used in

[1] Machine Learning in Science, University of Tübingen and Tübingen AI Center, Tübingen, Germany [2] Max Planck Institute for Intelligent Systems, Department Empirical Inference, Tübingen, Germany. Correspondence to: Manuel Gloeckler <manuel.gloeckler@uni-tuebingen.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

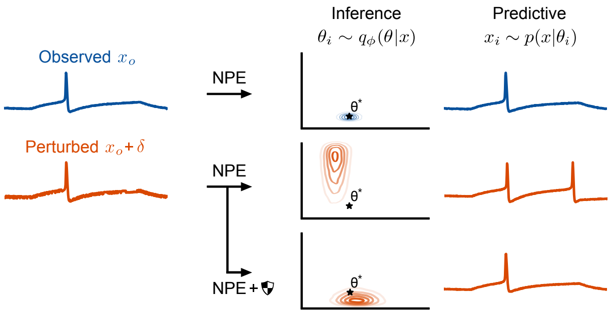

Figure 1. **Adversarial attack on amortized inference.** A minor perturbation of the observed data (left column, here: a voltage recording) creates a remarkably different estimate of the posterior over parameters (middle column, here: over parameters of a biophysical neuron model). Predictive samples from the adversarial posterior estimate are very different *both* from the observed and perturbed data (right column)–in this case, they exhibit two "spikes", while the original data only has a single one—showing that the attack leads to a break-down of the posterior *estimate* of the inference network (rather than a change in the true posterior). Our defense strategy leads to a more reliable posterior estimate (bottom row) with realistic predictive samples.

real-world safety-critical applications such as, e.g., robotics (Ramos et al., 2019) or applications accessible to the general public (Moon et al., 2023; Shen et al., 2023). In science and engineering, users are usually domain experts, but they are often not machine learning experts and, hence, must be aware of the limitations and brittleness of any such methods.

Here, we investigate the impact of adversarial attacks on amortized inference, focusing on a particular method for amortized Bayesian inference, namely Neural Posterior Estimation (NPE, Cranmer et al. 2020). While adversarial attacks have been extensively studied in the context of classification (Rauber et al., 2017; Croce et al., 2021; Li et al., 2022), we present an approach and benchmark problems for evaluating the adversarial robustness of neural networks approximating multi-dimensional Bayesian posterior distributions. Using this approach, we demonstrate that NPE can be highly vulnerable to adversarial attacks. Finally, we develop a computationally efficient method for improving the adversarial robustness of NPE, and demonstrate its utility on a real-world example from neuroscience.

Our overall approach is the following (Fig. 1): Given an observation $\boldsymbol{x}_o$, we consider an adversarial perturbation (Sec. 3.1). As we will show, even barely visible adversarial perturbations can strongly change estimated posterior distributions, and lead to predictive samples which strongly deviate from the original observation. We suggest and implement a defense strategy (Sec. 3.2), and will show that it reduces the impact on the posterior estimate, in particular, such that it still contains the ground truth parameters.

2. Background and Notation

2.1. Amortized Bayesian inference

In this work, we consider a fixed generative model that defines a relationship between $\boldsymbol{x}$ and unknown parameters $\boldsymbol{\theta}$, given by $p(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. By Bayes theorem, there exists a function $f : \mathcal{X} \rightarrow \mathcal{P}(\boldsymbol{\theta})$ which maps data onto the posterior distribution $f(\boldsymbol{x}_o) = p(\boldsymbol{\theta}|\boldsymbol{x}_o)$. As opposed to computing the posterior distribution for every observation, amortized Bayesian inference targets to learn the mapping f directly, thereby amortizing the cost of inference.

One method to perform amortized Bayesian inference is Neural Posterior Estimation (NPE). NPE first draws samples from the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{x})$ and then trains a conditional density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ with learnable parameters ϕ to approximate the posterior distribution:

$$\mathcal{L}(\phi) = \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x})\right] \approx \frac{1}{N}\sum_{i=1}^{N} -\log q_\phi(\boldsymbol{\theta}_i|\boldsymbol{x}_i)$$

If the conditional density estimator is sufficiently expressive, then this is minimized if and only if $q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) = p(\boldsymbol{\theta}|\boldsymbol{x})$ for all $\boldsymbol{x}$ in the support of $p(\boldsymbol{x})$ (Papamakarios & Murray, 2016).

2.2. Adversarial attacks and defenses

Szegedy et al. (2014) first proposed the concept of adversarial examples to fool neural networks. Adversarial examples are typically defined as solutions to an optimization problem (Szegedy et al., 2014; Goodfellow et al., 2015)

$$\tilde{\boldsymbol{x}} = \arg \max_{||\tilde{\boldsymbol{x}}-\boldsymbol{x}||_{\mathcal{X}} \leq \epsilon} \Delta(f(\tilde{\boldsymbol{x}}), f(\boldsymbol{x})),$$

where Δ specifies a distance between the predictions of the neural network.

Many defenses against adversarial examples have been proposed. We build upon a popular defense called TRADES (Zhang et al., 2019)– when translated to inference tasks, TRADES can be interpreted as regularizing the neural network loss with the Kullback-Leibler divergence between the clean data and an adversarially perturbed data point:

$$\begin{aligned}\mathcal{L}(\phi) =& \mathbb{E}_{p(\tilde{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{\theta})}[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) + \\ & \beta D_{KL}(q_\phi(\boldsymbol{\theta}|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}))]\end{aligned}$$

Here, $\tilde{\boldsymbol{x}}$ is obtained by generating an adversarial example *during training*. This regularization requires generating an adversarial example for every datapoint and epoch, which requires running several gradient descent steps for every datapoint $\boldsymbol{x}$– this would be exceedingly computationally costly for our inference tasks, but we will present methods for overcoming this limitation. To simplify notation, we abbreviate the posterior estimate given clean data as $q := q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ and given perturbed data as $\tilde{q} := q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}})$.

3. Methods

3.1. Adversarial attacks on amortized inference

Adversarial perturbations are typically studied in classification tasks, in which the perturbation makes the neural network predict a wrong class. For amortized Bayesian inference, however, the output of the neural network is a continuous probability distribution (the estimate of the posterior). We therefore define the target of the adversarial perturbation to maximize the divergence between the estimated posterior given the 'clean' vs. the adversarially perturbed data, i.e., $D_{KL}(q(\boldsymbol{\theta}|\boldsymbol{x})||q(\boldsymbol{\theta}|\boldsymbol{x}+\boldsymbol{\delta}))$ (Gondim-Ribeiro et al., 2018; Willetts et al., 2021; Dax et al., 2022; Dang-Nhu et al., 2020).

We here focus on the Kullback-Leibler divergence[1], but any divergence or pseudo-divergence (e.g. a distance function on moments of the posterior) would be possible (Gondim-Ribeiro et al., 2018; Willetts et al., 2021; Dax et al., 2022; Dang-Nhu et al., 2020). An attack is thus defined by the constrained optimization problem

$$\boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\delta}} D_{KL}\left(q_\phi(\boldsymbol{\theta}|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}|\boldsymbol{x}+\boldsymbol{\delta})\right) \text{ s.t. } ||\boldsymbol{\delta}|| \leq \epsilon.$$

To solve it, we use projected gradient descent (PGD) as an attacking scheme (Madry et al., 2018), following work on adversarial robustness for classification. We estimate the divergence between distributions parameterized by conditional normalizing flows using Monte Carlo sampling. We use the reparameterization trick (Kingma & Welling, 2014) to estimate gradients (details in A1.1).

We note that small perturbations to the observed data are expected to change the *true* posterior distribution. A sufficiently small perturbation will, in general, only cause a minor change in the posterior distribution (Latz, 2020). Furthermore, posterior predictive samples should match the perturbed observation (Berger et al., 1994; Sprungk, 2020). In contrast, we will demonstrate that the estimated posterior will change strongly after minor changes to the data, and that predictive samples of the posterior estimate do not match the perturbed observation, implying that the attack indeed breaks the amortized posterior *estimate*.

3.2. An adversarial defense for amortized inference

How do we modify NPE to be robust against such attacks? As described in Sec. 2.2, many adversarial defenses (e.g., TRADES) rely on generating adversarial examples during training, which can be computationally costly. In particular, for expressive conditional density estimators such as normalizing flows, generating an adversarial attack requires several Monte Carlo (MC) samples at every gradient step, thus rendering this approach exceedingly costly. Here, we propose a computationally efficient method based on a moving average estimate of the trace of the Fisher information matrix.

Regularizing by the Fisher information matrix To avoid having to generate adversarial examples during training, we exploit the fact that adversarial perturbations tend to be small and apply a second-order Taylor approximation to the KL-divergence (as has been done in previous work, Zhao et al. 2019; Shen et al. 2019; Miyato et al. 2016). This results in a quadratic expression (Blyth, 1994),

$$D_{KL}(q_\phi(\boldsymbol{\theta}|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}|\boldsymbol{x}+\boldsymbol{\delta})) \approx \frac{1}{2}\boldsymbol{\delta}^T\mathcal{I}_{\boldsymbol{x}}\boldsymbol{\delta},$$

where $\mathcal{I}_{\boldsymbol{x}}$ is the Fisher information matrix (FIM) with respect to $\boldsymbol{x}$, which is given by

$$\mathcal{I}_{\boldsymbol{x}} = \mathbb{E}_{q_\phi(\boldsymbol{\theta}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{x}}\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x})(\nabla_{\boldsymbol{x}}\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x}))^T\right].$$

This suggests that the neural network is most brittle along the eigenvector of the FIM with the largest eigenvalue (in particular, for a linear Gaussian model, the optimal attack on D_{KL} corresponds *exactly* to the largest eigenvalue of the FIM, Sec. A5).). To improve robustness along this direction, one can regularize with the largest eigenvalue of the FIM λ_{max} (Zhao et al., 2019; Shen et al., 2019; Miyato et al., 2016):

$$\mathcal{L}(\phi) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) + \beta\lambda_{\text{max}}\right].$$

While this approach overcomes the need to generate adversarial examples during training, computing the largest eigenvalue of the FIM can still be costly: First, it requires estimating an expectation over $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ to obtain the FIM and, second, computing the largest eigenvalue of a potentially large matrix. Below, we address these challenges.

Reducing the number of MC samples with moving averages For expressive density estimators such as normalizing flows, the expectation over $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ cannot be computed analytically, and has to be estimated with MC sampling:

$$\hat{\mathcal{I}}_{\boldsymbol{x}} = \frac{1}{N}\sum_i\left[\nabla_{\boldsymbol{x}}\log q_\phi(\boldsymbol{\theta}_i|\boldsymbol{x})(\nabla_{\boldsymbol{x}}\log q_\phi(\boldsymbol{\theta}_i|\boldsymbol{x}))^T\right]$$

To reduce the number of samples required, we exploit that consecutive training iterations result in small changes of the neural network, and use an exponential moving average estimator for the FIM, i.e., $\hat{\mathcal{I}}_{\boldsymbol{x}}^{(t)} = \gamma\hat{\mathcal{I}}_{\boldsymbol{x}} + (1-\gamma)\hat{\mathcal{I}}_{\boldsymbol{x}}^{(t-1)}$, where the superscript (t) indicates the training iteration.

Using the trace of the Fisher information matrix as regularizer Such an exponential moving average estimator

[1] We focus on $D_{KL}(q||\tilde{q})$ to generate and evaluate attacks, but we discuss and evaluate the effect of a different adversarial objective in Sec. A4.2.

Algorithm 1 FIM-regularized NPE

```
Inputs: conditional density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ with learnable parameters $\phi$, batch size $B$, number of training steps $T$, learning rate $\alpha$, regularization strength $\beta$, regularization momentum $\gamma$, number of Monte Carlo samples $N$
Initialize: $g^{(0)} = 0$
for $t = 1$ to $T$ do
    for $b = 1$ to $B$ do
        $\mathcal{L}(\phi) = -\frac{1}{B} \log q_\phi(\boldsymbol{\theta}_b|\boldsymbol{x}_b)$ // NPE loss
        $\boldsymbol{\theta}_{1b}, \dots, \boldsymbol{\theta}_{Nb} \sim q_\phi(\boldsymbol{\theta}|\boldsymbol{x}_b)$ // Monte Carlo
    end for
    $r = \frac{1}{B} \sum_b \frac{1}{N} \sum_i \sum_d \left[\nabla_{\boldsymbol{x}} \log q_\phi(\boldsymbol{\theta}_{ib}|\boldsymbol{x}_b)\right]_d^2$
                                        // FIM Trace
    $g^{(t)} = \gamma \nabla_\phi r + (1-\gamma) g^{(t-1)}$ // moving average
    $\phi^t = \phi^{t-1} - \alpha(\text{ADAM}(\nabla_\phi \mathcal{L}(\phi) + \beta g^{(t)}))$
end for
```

decreases the number of required MC samples, but it would require storing the FIM for each $\boldsymbol{x}$ and computing the FIM's largest eigenvalue at every iteration. Computing the largest eigenvalue scales cubically with the number of dimensions of $\boldsymbol{x}$ (but could be scaled with power-iterations, Miyato et al. 2016) and obtaining the largest eigenvalue of a random matrix (such as the MC-estimated FIM) requires many MC samples (Hayashi et al., 2018; Hayou, 2017). To overcome these limitations, we regularize instead with the trace of the FIM, which is an upper bound to the largest eigenvalue.

Unlike the largest eigenvalue, the trace of the FIM can be computed from MC samples quickly and without explicitly computing the FIM. Using the trace of the FIM simplifies the moving average estimator to

$$\text{tr}(\hat{\mathcal{I}}_{\boldsymbol{x}}^{(t)}) = \gamma \text{tr}(\hat{\mathcal{I}}_{\boldsymbol{x}}) + (1-\gamma)\text{tr}(\hat{\mathcal{I}}_{\boldsymbol{x}}^{(t-1)}).$$

To avoid maintaining the computation graph for every $\boldsymbol{x}$ and (t), we store the *average* gradient with respect to the neural network parameters instead of storing $\text{tr}(\hat{\mathcal{I}}_{\boldsymbol{x}}^{(t)})$ directly,

$$g^{(t)} := \nabla_\phi \mathbb{E}_{p(\boldsymbol{x})}\left[\text{tr}(\hat{\mathcal{I}}_{\boldsymbol{x}}^{(t)})\right].$$

Summary and illustration Our adversarial defense is summarized in Algorithm 1. At every iteration, the method computes the Monte Carlo average of the trace of the Fisher information, updates the moving average of this quantity, and uses it as a regularizer to the negative log-likelihood loss. Despite our approximations, our method performs similarly to regularizers based on the largest eigenvalue or trace of the exact FIM (comparison with a Gaussian density estimator on the VAE task in Sec. A7). Finally, we note that using the FIM-regularizer systematically changes the posterior estimate even with infinite training data and, therefore, leads to a trade-off between accuracy on clean data and robustness to perturbations (Sec. A3). For a generalized linear Gaussian density estimator, the bias induced by FIM-regularization can be calculated exactly (details in Sec. A6).

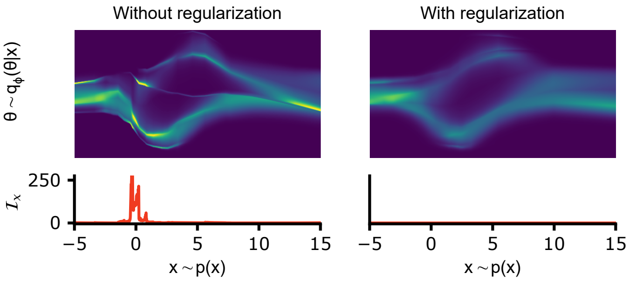

Figure 2. **Regularizing conditional density estimators by the Fisher information matrix (FIM).** We trained a neural spline flow to estimate a conditional density with negative log-likelihood loss (left) and with our FIM regularizer (right). The Fisher information (bottom) is high in regions that are non-smooth along the conditioning variable x. The regularized loss leads to density estimates which are smoother while still being able to capture complex densities.

We demonstrate the method on a simple one-dimensional conditional density estimation task using a neural spline flow (Dolatabadi et al., 2020) (Fig. 2). The Fisher information is large in $\boldsymbol{x}$-regions where $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ changes quickly as a function of $\boldsymbol{x}$. By regularizing with the trace of the FIM, the learned density is significantly smoother.

4. Experimental results

4.1. Benchmark tasks

We first evaluated the robustness of Neural Posterior Estimation (NPE) and the effect of FIM-regularization on six benchmark tasks (details in Sec. A1.2). Rather than using established benchmark tasks (Lueckmann et al., 2021), we chose tasks with more high-dimensional data, which might offer more flexibility for adversarial attacks.

Visualizing adversarial attacks We first visualized the effect of several adversarial examples on inference models trained with standard (i.e., unregularized) NPE. We trained NPE with a Masked Autoregressive Flow (MAF, Papamakarios et al. 2017) on $100k$ simulations and generated an adversarial attack for a held-out datapoint. Although the perturbations to the observations are hardly perceptible, the posterior estimates change drastically, and posterior predictive samples match neither the clean nor the perturbed observation (Fig. 3). This indicates that the attacked density estimator predicts a posterior distribution that does not match the true Bayesian posterior given the perturbed datapoint $p(\boldsymbol{\theta}|\tilde{\boldsymbol{x}})$, but rather it predicts an incorrect distribution.

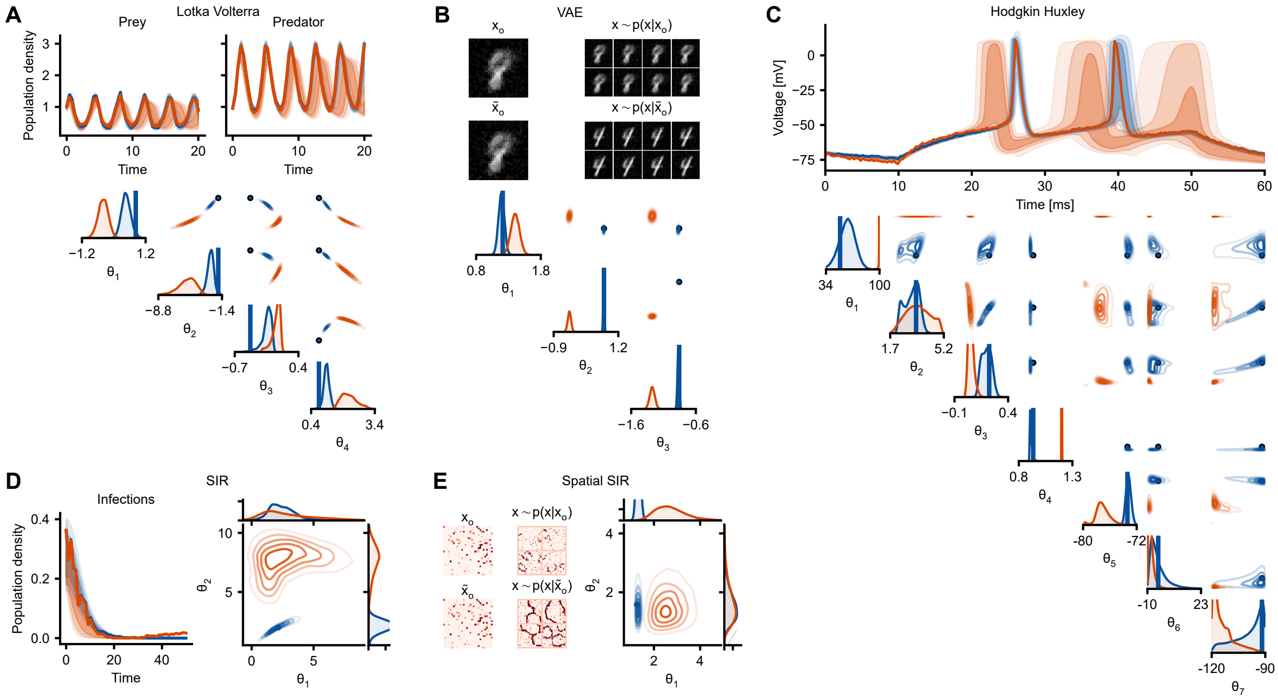

Figure 3. **Adversarial examples for each benchmark task**. Each panel shows i) the original observation (blue line) and corresponding posterior predictive samples (blue shaded), ii) the adversarial example (orange line) and posterior predictive samples based on the perturbed posterior estimate, and iii) posterior distribution plots with the posterior estimate for the original (blue) and perturbed (orange) data, and the ground-truth parameters (black dot).

How does the adversarial attack change the prediction of the neural density estimator so strongly? We investigated two possibilities for this: First, the adversarial attack could construct a datapoint $\tilde{x}$ which is misspecified. Previous work has reported that NPE can perform poorly in the presence of misspecification (Cannon et al., 2022). Indeed, on the SIR benchmark task (Fig. 3D), we find clues that are consistent with misspecification: At the end of the simulation ($t > 20$), the perturbed observation shows an increase in infections although they had already nearly reached zero. Such an increase cannot be modeled by the simulator and cannot be attributed to the noise model (since the noise is log-normal and, thus, small for low infection counts).

A second possibility for the adversarial attack to strongly change the posterior estimate would be to exploit the neural network itself and generate an attack for which the network produces poor predictions. We hypothesized that, on our benchmark tasks, this possibility would dominate. To investigate this, we performed adversarial attacks on different density estimators and evaluated how similar the adversarial attacks were to each other (Fig. A3). We find that the attacks largely differ between different density estimators, suggesting that the attacks are indeed targeted to the specific neural network.

Quantifying the impact of adversarial attacks We quantified the effect of adversarial attacks on NPE without using an adversarial defense. After training NPE with 100k simulations, we constructed adversarial attacks for 10^4 held-out datapoints (as described in Sec. 3.1). As a baseline, we also added a random perturbation of the same magnitude on each datapoint. We then computed the average D_{KL} between the posterior estimates given clean and perturbed data (Fig. 4). For all tasks and tolerance levels (the scale of the perturbation), the adversarial attack increases the D_{KL} more strongly than a random attack. In addition, for all tasks apart from the linear Gaussian task, the difference between the adversarial and the random attack is several orders of magnitude (Fig. 4A).

As a second evaluation-metric, we computed the expected coverage of the perturbed posterior estimates, which allows us to study whether posterior estimates are under-, or overconfident (Fig. 4B, details in Sec. A1.3) (Cannon et al., 2022). For stronger perturbations, the posterior estimates become overconfident around wrong parameter regions and show poor coverage. As expected, adversarial attacks impact the coverage substantially more strongly than random attacks.

Additional results for different density estimators, alternative attack definitions, and simulation budgets can be found

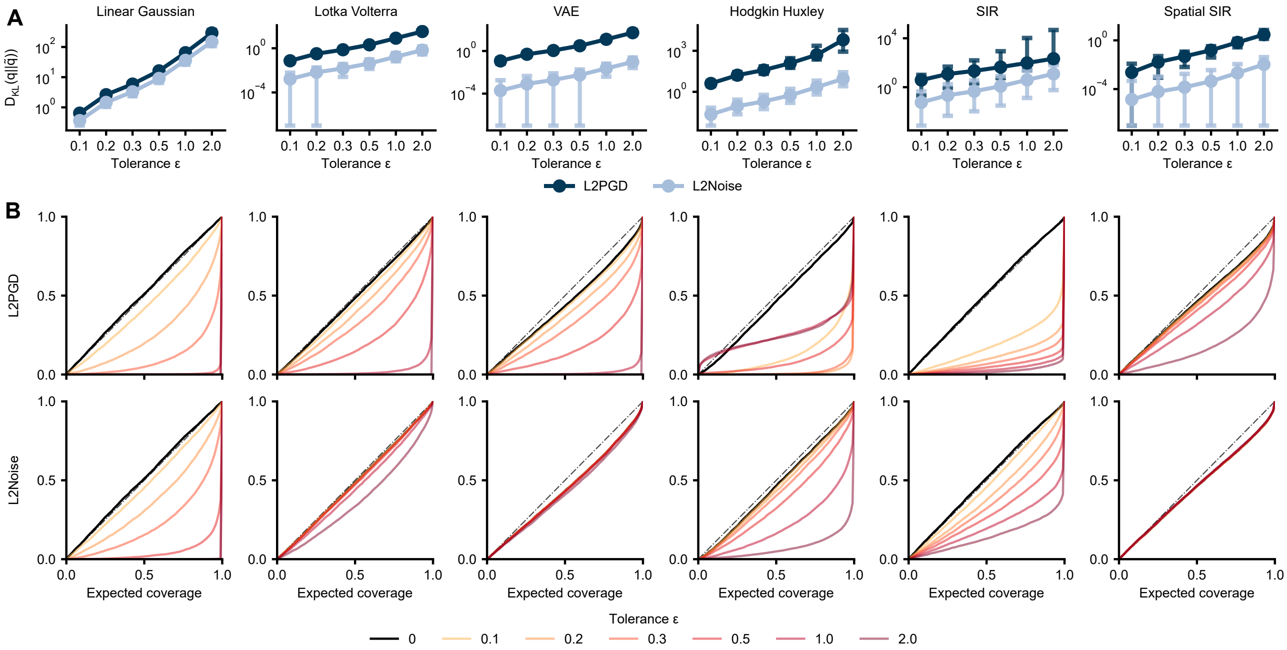

Figure 4. **Adversarial attacks on neural posterior estimation**. **(A)** KL-divergence between posterior estimates for original and perturbed data, $D_{KL}(q||\tilde{q})$ for targeted (L2PGD) and random (L2Noise) attacks on a linear Gaussian model and six benchmark tasks (details in Sec. A1.2), for several tolerance levels. Error bars show 15% and 85% quantiles. **(B)** Nominal coverage vs. empirical expected coverage for L2PGD (top) and L2Noise (bottom) attacks. The dotted line is identity.

in Sec. A4.2 (Figs. A4, A1, A5). The results are mostly consistent across different density estimators (with minor exceptions at low simulation budgets), indicating that more flexible estimators are not necessarily less robust.

Adversarial defense of NPE Next, we evaluated the adversarial robustness when regularizing NPE with the moving average estimate of the trace of the Fisher Information Matrix (FIM) (Sec. 3.2). In addition, we evaluated two approaches adapted from defense methods for classification tasks– however, both of these approaches rely on generating adversarial examples during training and are, thus, more computationally expensive (details in Sec. A2, methods are labeled as ‘Adv. training’ and ‘TRADES’).

All adversarial defense methods significantly reduce the ability of attacks to change the posterior estimate (Fig. 5A). In addition, the FIM regularizer performs similarly to other defense methods but is computationally much more efficient and scalable (A4.1, Fig. A2, sweeps for β in Fig. A6).

We evaluated the expected coverage when using FIM regularization (Fig. 5B, results for Adv. Training and TRADES in Fig. A7). For all tasks, the coverage is shifted towards the upper left corner, indicating a more conservative posterior estimate (further analysis in Sec. A3). Even for medium to high tolerance levels (i.e., strong perturbations), the posterior estimate often remains underconfident and covers the true parameter set, a behavior which has been argued to be desirable in scientific applications (Hermans et al., 2022). Other defense methods (that were not specifically developed as adversarial defenses), such as posterior ensembles or noise augmentation, barely increase the adversarial robustness of NPE (Fig. A7, Sec. A4.3). Further, we investigate this effect directly comparing against the true posterior (as estimated via MCMC for a subset of tasks) in Sec. A8, verifying that posterior approximation on adversarial perturbed data is poor but can be improved using FIM regularization.

Finally, we studied the trade-off between robustness to adversarial perturbations and accuracy of the posterior estimate on unperturbed data (Zhang et al., 2019; Tsipras et al., 2019). We computed the accuracy on unperturbed data (evaluated as average log-likelihood) and the robustness to adversarial perturbations (measured as D_{KL} between clean and perturbed posteriors) for a range of regularization strengths β (Fig. 5C). For a set of intermediate values for β, it is possible to achieve a large gain in robustness while only weakly reducing accuracy (details in Sec. A3, results for other density estimators in Sec. A4.3, Figs. A6 and A8).

Overall, FIM regularization is a computationally efficient method to reduce the impact of adversarial examples on NPE. While it encourages underconfident posterior estimates, it allows for high robustness with a relatively modest reduction in accuracy.

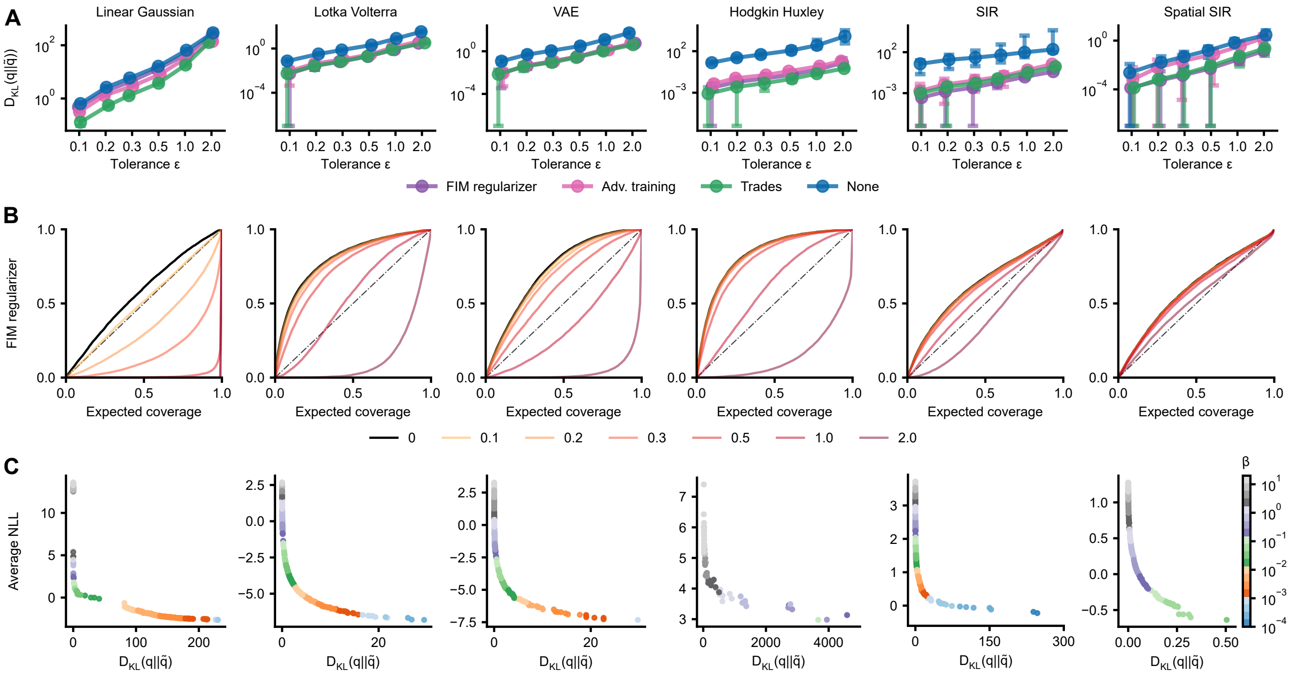

Figure 5. **Defenses against adversarial attacks**. **(A)** KL-divergences $D_{KL}(q||\tilde{q})$ for all three defenses (FIM regularisation, adversarial training and TRADES, see A4.3 for details) and without defense for six benchmark tasks. **(B)** Expected coverage for FIM regularization. **(C)** Trade-offs between accuracy (average of log-likelihood on unperturbed data) and robustness ($D_{KL}(q||\tilde{q})$) for $\epsilon = 2.0$. For a range of regularisation strengths β, a large gain in robustness only leads to a small drop in accuracy.

4.2. Neuroscience example: Pyloric network

Finally, we performed adversarial attacks and defenses on a real-world simulator of the pyloric network in the stomatogastric ganglion (STG) of the crab *Cancer Borealis*. The simulator includes three model neurons, each with eight membrane conductances and seven synapses (31 parameters in total) (Prinz et al., 2003; 2004). Prior studies have used extensive simulations from prior samples and performed amortized inference with NPE (18 million simulations in Gonçalves et al. (2020), 9 million in Deistler et al. (2022b)). Both of these studies used hand-crafted summary statistics. In contrast, we here performed inference on the raw traces (subsampled by a factor of 100 due to memory constraints).

After subsampling, the data contains three voltage traces, each of length 800. We ran 8M simulations from the prior and excluded any parameters which generated physiologically implausible data (Lueckmann et al., 2017), resulting in a dataset with $750k$ datapoints. We used a one-dimensional convolutional neural network for each of the three traces and passed the resulting embedding through a fully-connected neural network (Fig. 6).

The neural density estimator, trained without regularization, is susceptible to adversarial attacks (Fig. 6A). Given unperturbed data, the posterior predictive closely matches the data, whereas for the adversarially perturbed data the posterior predictive no longer matches the observations. In addition, the predicted posterior given clean data strongly differs from the predicted posterior given adversarially perturbed data.

In contrast, when regularizing with the FIM approach, the neural density estimator becomes significantly more robust to adversarial perturbations (Fig. 6B). The posterior predictives now closely matched the data, both for clean as well as adversarially perturbed observations. In addition, the posterior estimates given clean and perturbed observations match closely.

We quantified these results by computing the D_{KL} between clean and adversarially perturbed posterior estimates as well as the expected coverage (Fig. 6C). NPE without regularization has a higher D_{KL} and is overconfident. In contrast, the FIM-regularized posterior is underconfident, even for strong perturbations. These results demonstrate that real-world simulators can strongly suffer from adversarial attacks. The results also show that our proposed FIM-regularizer scales to challenging and high-dimensional tasks.

5. Discussion

We showed that amortized Bayesian inference can be vulnerable to adversarial attacks. The posterior estimate can change strongly when slightly perturbing the observed data,

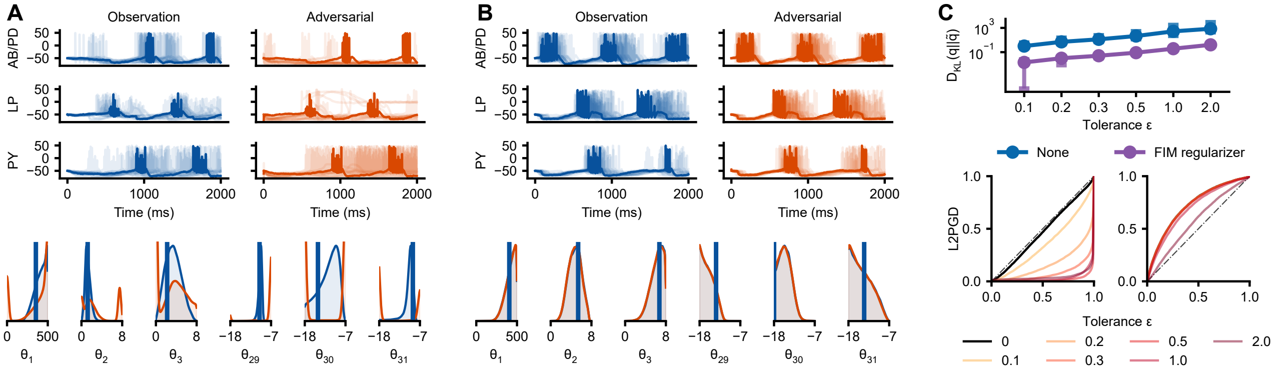

Figure 6. **Attack and defense on pyloric network simulator. (A)** Adversarial attack on NPE. Top: Observation (blue line), posterior predictives (blue shades), adversarially perturbed observation (orange line) and corresponding posterior predictives (orange shades). Bottom: Subset of marginals of posterior distribution given clean (blue) and perturbed (orange) data. Blue line is the true parameter set. **(B)** Same as A, but when employing FIM-regularization. **(C)** Top: $D_{KL}(q||\tilde{q})$ for NPE and FIM-regularized NPE. Bottom: Expected coverage for NPE (left) and FIM-regularized NPE (right).

leading to inaccurate inference results. This poses a difficult challenge for amortized Bayesian inference which would severely limit its utility for applications in which trustworthy posterior estimates are essential: If small changes in the input data can have a strong impact on inference results, using misspecified data, or simply data not encountered during training could also lead to severely wrong conclusions.

To address this issue, we propose a computationally efficient defense strategy that can be used to reduce the vulnerability of Neural Posterior Estimation to adversarial attacks. We demonstrate the effectiveness of this method and show that it can significantly improve the robustness and reliability of NPE in the presence of adversarial attacks.

Prior work on adversarial attacks and defenses Adversarial attacks and defenses have been studied on variational autoencoders (Kuzina et al., 2022; Husain & Knoblauch, 2022; Shu et al., 2018; Barrett et al., 2022; Willetts et al., 2021; Akrami et al., 2022). Our work differs from these papers in that we focus on posterior distributions parameterized by expressive conditional density estimators such as normalizing flows. Note, crucially, that the attacks in this context are on the conditioning-variable, in contrast to previous work on flow-based models studying attacks on the *output* of (unconditional) flow-based models (Pope et al., 2020). To train our inference model, we use the negative log-likelihood as loss-function (as compared to the ELBO for variational autoencoders), which makes our approach applicable to non-differentiable and implicit models. Several recent studies have proposed improvements to the robustness of NPE (Dellaporta et al., 2022; Lemos et al., 2022; Ward et al., 2022; Matsubara et al., 2022; Finlay & Oberman, 2019), but none of them have considered defenses against adversarial attacks. Dax et al. (2022) proposed the use of (likelihood-based) importance sampling to identify and correct poor approximations in an application from astrophysics, and in this context also evaluated the adversarial robustness of a neural posterior estimator. Concurrent theoretical work of Altekrüger et al. (2023) established basic conditions under which conditional density estimators on convergence are provably robust in particular depending on the Lipschitz constant of the inference network with respect to the observation.

Amortized Bayesian inference We studied adversarial robustness of a particular amortized Bayesian inference algorithm, Neural Posterior Estimation (NPE). Other simulation-based inference methods can be categorized as amortized as well, e.g., Neural Ratio Estimation (NRE) or Neural Likelihood Estimation (NLE) (Cranmer et al., 2020; Hermans et al., 2020; Papamakarios et al., 2019). These methods do not require new simulations or network training for new observations, but they require a (potentially expensive) inference phase to obtain the posterior. In addition, another approach to amortized Bayesian inference would be to perform amortized variational inference, which requires a differentiable model and likelihood-evaluations. We leave the study of adversarial attacks in these methods to future work.

Model misspecification and adversarial robustness Previous work (Cannon et al., 2022) raised concerns about the reliability of NPE on misspecified simulators. Adversarial examples exploit the brittleness of neural networks to construct examples on which NPE performs particularly poorly. As such, our study can be considered as a worst-case scenario of how minor deviations in the observed data can impact its reliability. We find that adversarial examples depend strongly on the network (for the same simulator), indicating the crucial role of the inference network.

Limitations Using a defense against adversarial attacks comes at two costs: Increased computational cost and, potentially, broader posteriors on clean data. Our proposed regularization scheme largely reduces computational cost (by up to an order of magnitude compared to other defense methods such as TRADES), but it, nonetheless, requires drawing Monte Carlo samples from the posterior estimate and evaluating its gradient w.r.t. every datapoint at every epoch. Across six benchmark tasks, our regularizer increased training time by a factor of four (compared to standard NPE, Fig. A2). Our analysis places emphasis on inference networks that have the capability to learn summary statistics of complex data, if necessary, in an end-to-end manner using neural networks (Chan et al., 2018; Radev et al., 2020). However, in various applications, expert-crafted summary statistics are commonly employed which can be explicitly designed to be robust against certain perturbations.

It has been argued that, in many applications (and in particular in the natural sciences), it is desirable to have underconfident posteriors (Hermans et al., 2022) – however, posterior estimates that are systematically too broad lead to a lower rate of learning from data and, thus, slower information acquisition. While we demonstrated that our method is comparable to NPE in terms of negative log-likelihood, users might need to evaluate the trade-off between robustness and information acquisition for their applications.

Acknowledgements

We thank Pedro Goncalves, Jaivardhan Kapoor, and Maximilian Dax for discussions and comments on the manuscript. MG and MD are supported by the International Max Planck Research School for Intelligent Systems (IMPRS-IS). This work was supported by the German Research Foundation (DFG) through Germany's Excellence Strategy – EXC-Number 2064/1 – Project number 390727645, the German Federal Ministry of Education and Research (Tübingen AI Center, FKZ: 01IS18039A), and the 'Certification and Foundations of Safe Machine Learning Systems in Healthcare' project funded by the Carl Zeiss Foundation.

Software and Data

We used PyTorch for all neural networks (Paszke et al., 2019) and hydra to track all configurations (Yadan, 2019). Code to reproduce results is available at `https://github.com/mackelab/RABI`.

References

Akrami, H., Joshi, A. A., Li, J., Aydöre, S., and Leahy, R. M. A robust variational autoencoder using beta divergence. *Knowledge-Based Systems*, 238:107886, 2022. Publisher: Elsevier.

Altekrüger, F., Hagemann, P., and Steidl, G. Conditional generative models are provably robust: Pointwise guarantees for bayesian inverse problems. *arXiv preprint arXiv:2303.15845*, 2023.

Andrieu, C. and Thoms, J. A tutorial on adaptive mcmc. *Statistics and computing*, 18:343–373, 2008.

Arnst, M., Louppe, G., Van Hulle, R., Gillet, L., Bureau, F., and Denoël, V. A hybrid stochastic model and its Bayesian identification for infectious disease screening in a university campus with application to massive COVID-19 screening at the University of Liège. *Mathematical Biosciences*, 347:108805, May 2022.

Barrett, B., Camuto, A., Willetts, M., and Rainforth, T. Certifiably robust variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pp. 3663–3683. PMLR, 2022.

Berger, J. O., Moreno, E., Pericchi, L. R., Bayarri, M. J., Bernardo, J. M., Cano, J. A., De la Horra, J., Martín, J., Ríos-Insúa, D., Betrò, B., Dasgupta, A., Gustafson, P., Wasserman, L., Kadane, J. B., Srinivasan, C., Lavine, M., O'Hagan, A., Polasek, W., Robert, C. P., Goutis, C., Ruggeri, F., Salinetti, G., and Sivaganesan, S. An overview of robust Bayesian analysis. *Test*, 3(1):5–124, June 1994.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 20(28):1–6, 2019. ISSN 1533-7928.

Blyth, S. Local Divergence and Association. *Biometrika*, 81 (3):579–584, 1994. Publisher: [Oxford University Press, Biometrika Trust].

Boelts, J., Lueckmann, J.-M., Gao, R., and Macke, J. H. Flexible and efficient simulation-based inference for models of decision-making. *eLife*, 11:e77220, July 2022.

Cannon, P., Ward, D., and Schmon, S. M. Investigating the Impact of Model Misspecification in Neural Simulation-based Inference, September 2022. arXiv:2209.01845 [cs, stat].

Chan, J., Perrone, V., Spence, J. P., Jenkins, P. A., Mathieson, S., and Song, Y. S. A Likelihood-Free inference framework for population genetic data using exchangeable neural networks. *Adv Neural Inf Process Syst*, 31: 8594–8605, December 2018.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Dang-Nhu, R., Singh, G., Bielik, P., and Vechev, M. Adversarial attacks on probabilistic autoregressive forecasting models. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2356–2365. PMLR, 13–18 Jul 2020.

Dax, M., Green, S. R., Gair, J., Macke, J. H., Buonanno, A., and Schölkopf, B. Real-time gravitational wave science with neural posterior estimation. *Physical review letters*, 127(24):241103, 2021.

Dax, M., Green, S. R., Gair, J., Pürrer, M., Wildberger, J., Macke, J. H., Buonanno, A., and Schölkopf, B. Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference, October 2022. arXiv:2210.05686 [astro-ph, physics:gr-qc].

Deistler, M., Goncalves, P. J., and Macke, J. H. Truncated proposals for scalable and hassle-free simulation-based inference. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a.

Deistler, M., Macke, J. H., and Gonçalves, P. J. Energy-efficient network activity from disparate circuit parameters. *Proceedings of the National Academy of Sciences*, 119(44):e2207632119, 2022b.

Dellaporta, C., Knoblauch, J., Damoulas, T., and Briol, F.-X. Robust Bayesian Inference for Simulator-based Models via the MMD Posterior Bootstrap. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 943–970. PMLR, May 2022. ISSN: 2640-3498.

Dolatabadi, H. M., Erfani, S., and Leckie, C. Invertible generative modeling using linear rational splines. In *International Conference on Artificial Intelligence and Statistics*, pp. 4236–4246. PMLR, 2020.

Finlay, C. and Oberman, A. M. Scaleable input gradient regularization for adversarial robustness, October 2019. arXiv:1905.11468 [cs, stat].

Golub, G. H., Hansen, P. C., and O'Leary, D. P. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21(1):185–194, 1999.

Gondim-Ribeiro, G., Tabacof, P., and Valle, E. Adversarial attacks on variational autoencoders. *arXiv preprint arXiv:1806.04646*, 2018.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., and Vogels, T. P. Training deep neural density estimators to identify mechanistic models of neural dynamics. *Elife*, 9:e56261, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*, 2015.

Greenberg, D., Nonnenmacher, M., and Macke, J. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pp. 2404–2414. PMLR, 2019.

Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., and Sriperumbudur, B. K. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Grünwald, P. and Ommen, T. v. Inconsistency of Bayesian Inference for Misspecified Linear Models, and a Proposal for Repairing It. *Bayesian Analysis*, 12(4):1069–1103, December 2017. Publisher: International Society for Bayesian Analysis.

Hayashi, K., Yuan, K.-H., and Liang, L. On the Bias in Eigenvalues of Sample Covariance Matrix. In Wiberg, M., Culpepper, S., Janssen, R., González, J., and Molenaar, D. (eds.), *Quantitative Psychology*, Springer Proceedings in Mathematics & Statistics, pp. 221–233, Cham, 2018. Springer International Publishing.

Hayou, S. On the overestimation of the largest eigenvalue of a covariance matrix, August 2017. arXiv:1708.03551 [math, q-fin, stat].

Hermans, J., Begy, V., and Louppe, G. Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pp. 4239–4248. PMLR, 2020.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., Begy, V., and Louppe, G. A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful, December 2022. arXiv:2110.06581 [cs, stat].

Holden, L., Hauge, R., and Holden, M. Adaptive independent Metropolis–Hastings. *The Annals of Applied Probability*, 19(1):395 – 413, 2009. doi: 10.1214/08-AAP545.

Husain, H. and Knoblauch, J. Adversarial Interpretation of Bayesian Inference. In *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, pp. 553–572. PMLR, March 2022. ISSN: 2640-3498.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

Kuzina, A., Welling, M., and Tomczak, J. M. Alleviating adversarial attacks on variational autoencoders with MCMC. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Latz, J. On the well-posedness of Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 8(1): 451–482, January 2020. arXiv:1902.10257 [cs, math, stat].

Lemos, P., Cranmer, M., Abidi, M., Hahn, C., Eickenberg, M., Massara, E., Yallup, D., and Ho, S. Robust Simulation-Based Inference in Cosmology with Bayesian Neural Networks. In *ICML 2022 Workshop on Machine Learning for Astrophysics*, 2022. arXiv:2207.08435 [astro-ph].

Li, Y., Cheng, M., Hsieh, C.-J., and Lee, T. C. M. A Review of Adversarial Attack and Defense for Classification Methods. *The American Statistician*, 76(4):329–345, October 2022. arXiv:2111.09961 [cs].

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30, 2017.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 343–351. PMLR, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Matsubara, T., Knoblauch, J., Briol, F.-X., and Oates, C. J. Robust Generalised Bayesian Inference for Intractable Likelihoods, January 2022. arXiv:2104.07359 [math, stat].

Medina, M. A., Olea, J. L. M., Rush, C., and Velez, A. On the robustness to misspecification of Î±-posteriors and their variational approximations. *Journal of Machine Learning Research*, 23(147):1–51, 2022.

Min, Y., Chen, L., and Karbasi, A. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 129–139. PMLR, 27–30 Jul 2021.

Miyato, T., Maeda, S.-i., Koyama, M., Nakae, K., and Ishii, S. Distributional Smoothing with Virtual Adversarial Training. In *International Conference on Learning Representations*, 2016.

Moon, H.-S., Oulasvirta, A., and Lee, B. Amortized inference with user simulations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–20, 2023.

Neal, R. M. Slice sampling. *The annals of statistics*, 31(3): 705–767, 2003.

Papamakarios, G. and Murray, I. Fast ϵ-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Papamakarios, G., Sterratt, D., and Murray, I. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 837–848. PMLR, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. arXiv:1912.01703 [cs, stat].

Pope, P., Balaji, Y., and Feizi, S. Adversarial robustness of flow-based generative models. In *International Conference on Artificial Intelligence and Statistics*, pp. 3795–3805. PMLR, 2020.

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., and Destexhe, A. Minimal Hodgkin–Huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99(4):427–441, November 2008.

Prinz, A. A., Billimoria, C. P., and Marder, E. Alternative to hand-tuning conductance-based models: construction and analysis of databases of model neurons. *Journal of neurophysiology*, 2003.

Prinz, A. A., Bucher, D., and Marder, E. Similar network activity from disparate circuit parameters. *Nature Neuroscience*, 7(12):1345–1352, December 2004. Number: 12 Publisher: Nature Publishing Group.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Ramos, F., Possas, R. C., and Fox, D. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators, 2019.

Rauber, J., Brendel, W., and Bethge, M. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017.

Ribeiro, A. H., Zachariah, D., and Schön, T. B. Surprises in adversarially-trained linear regression. *arXiv preprint arXiv:2205.12695*, 2022.

Rozet, F. et al. Arbitrary marginal neural ratio estimation for likelihood-free inference. *arXiv preprint arXiv:2110.00449*, 2021.

Schmitt, M., Bürkner, P.-C., Köthe, U., and Radev, S. T. Detecting Model Misspecification in Amortized Bayesian Inference with Neural Networks, November 2022. arXiv:2112.08866 [cs, stat].

Shen, C., Peng, Y., Zhang, G., and Fan, J. Defending against adversarial attacks by suppressing the largest eigenvalue of fisher information matrix. *arXiv preprint arXiv:1909.06137*, 2019.

Shen, M., Ghosh, S., Sattigeri, P., Das, S., Bu, Y., and Wornell, G. Reliable gradient-free and likelihood-free prompt tuning. In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 2416–2429, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

Shu, R., Bui, H. H., Zhao, S., Kochenderfer, M. J., and Ermon, S. Amortized Inference Regularization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Sprungk, B. On the Local Lipschitz Stability of Bayesian Inverse Problems. *Inverse Problems*, 36(5):055015, May 2020. arXiv:1906.07120 [cs, math, stat].

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., and Macke, J. H. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020.

Trivedi, S. and Wang, J. The expected jacobian outerproduct: Theory and empirics, 2020.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

von Krause, M., Radev, S. T., and Voss, A. Mental speed is high until age 60 as revealed by analysis of over a million participants. *Nature human behaviour*, 6(5):700–708, 2022.

Vovk, V. G. Aggregating strategies. In *Proceedings of the third annual workshop on Computational learning theory*, COLT '90, pp. 371–386, San Francisco, CA, USA, July 1990. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-146-8.

Ward, D., Cannon, P., Beaumont, M., Fasiolo, M., and Schmon, S. M. Robust neural posterior estimation and statistical model criticism. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Willetts, M. J., Camuto, A., Rainforth, T., Roberts, S., and Holmes, C. C. Improving {vae}s' robustness to adversarial attack. In *International Conference on Learning Representations*, 2021.

Yadan, O. Hydra - a framework for elegantly configuring complex applications. Github, 2019.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Zhao, C., Fletcher, P. T., Yu, M., Peng, Y., Zhang, G., and Shen, C. The adversarial attack and detection under the fisher information metric. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5869–5876, 2019. Issue: 01.

Appendix

A1. Further experimental details

A1.1. Training procedure

All methods and evaluations were conducted using PyTorch (Paszke et al., 2019). We used Pyro (Bingham et al., 2019) implementations of masked autoregressive flow (MAF) or rational linear spline flow (NSF) (Papamakarios et al., 2017; Dolatabadi et al., 2020). We employed three transforms, each of which was parameterized using a two-layered ReLU multi-layer perceptron (MLP) with 100 hidden units. For higher dimensional tasks such as Hodgkin Huxley, VAE, and Spatial SIR, we used a ReLU MLP embedding network with 400, 200, and 100 hidden units and outputting a 50 dim vector.

The diagonal Gaussian model used two hidden layers, a hidden layer of size 100, and ReLU activations. The Multivariate Gaussian and mixture density networks used a hidden layer of size 200.

We trained each model with the Adam optimizer with a learning rate of 10^{-3}, a batch size of 512, and a maximum of 300 epochs. If training failed, the learning rate was reduced to 10^{-4} or 10^{-5}. To prevent overfitting, we used early stopping based on a validation loss evaluated on 512 hold-out samples. Each model was trained on the same set of either 10^3, 10^4, or 10^5 simulations.

For the adversarial attacks, we performed 200 projected gradient descent steps and estimated the D_{KL} with a Monte Carlo average of 5 samples at each step (for the Gaussian models we used the analytical solution). Additionally, each adversarial example was clamped to the minimum and maximum values within the test set to avoid generating samples outside of the support of $p(\boldsymbol{x})$. After the optimization was finished, for the evaluation of the attack, the adversarial objective was evaluated with a Monte Carlo budget of 256 samples. We note that, as the scale of data varies strongly between simulators, all tolerance levels ϵ were normalized by the average of the standard deviation of prior predictives. Table 1 shows the resulting tolerance levels.

For the results shown in Fig. 5, we used a different value for the regularization strength β for each task, as the parameter is coupled to the magnitude of the perturbation (which is different for each task). The particular value of β was hand-picked based on initial experiments. The values were: $\beta = 0.001$ for Gaussian linear, $\beta = 0.1$ for SIR, $\beta = 0.01$ for Lotka Volterra, $\beta = 100$ for Hodgkin Huxley, $\beta = 0.01$ for VAE and $\beta = 0.1$ for Spatial SIR. Sweeps for β for each task are shown in Fig. A6. We used 5 Monte Carlo samples per iteration and a momentum of $\gamma = 0.85$ for each benchmark task for the FIM regularizer. We used a MAF for all benchmark results in the main paper and evaluated different density estimators in Figs. A4 and A1.

For the pyloric network task, we employed a MAF with three transforms, each parameterized by a 3-layer neural network with 200 hidden neurons. We also utilized an embedding network composed of three 1D convolutional neural networks, each with three convolutional layers that produce six, nine, and twelve output channels. These networks were applied to the voltage trace of each neuron. The results were then summarized by a 3-layer feed-forward neural network, and reduced to a 100-dimensional feature vector. We trained the model using 750,000 pre-selected simulations and evaluated its convergence on a validation set of 4096 additional datapoints. The evaluation was conducted on 10,000 separate simulations. For the FIM regularized model, we used $\beta = 100$. Further, we set the number of Monte Carlo samples within the exponential moving average to one; the momentum remained $\gamma = 0.85$.

A1.2. Benchmark tasks

We used the following benchmark tasks, which produce relatively complex and high-dimensional data. We used these tasks instead of established benchmark tasks (Lueckmann et al., 2021) because our tasks are chosen to have more high-dimensional data and, thus, might offer more flexibility for adversarial attacks.

Gaussian Linear: A simple diagonal linear mapping (entries sampled from a standard Gaussian) of a ten-dimensional vector subject to isotropic Gaussian noise:

$$p(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}; \mathbf{A}\boldsymbol{\theta}, \sigma^2 \mathbf{I}) \mathcal{N}(\boldsymbol{\theta}, \mathbf{0}, \mathbf{I})$$

with $\sigma = 0.1$. As a result, the posterior is also Gaussian and analytically tractable.

Lotka Volterra: An ecological predator-prey model with four parameters. It is given by the solution of the following differential equation:

$$\frac{dx}{dt} = \alpha x - \beta xy \quad \frac{dy}{dt} = \delta xy - \gamma y$$

with α representing the growth rate of prey, β the death rate of prey, δ the hunting efficiency of the predator and γ the death rate of the predator. The observed data are the predator and prey population densities at 50 equally spaced time points. We added normally distributed noise with $\sigma = 0.05$. The prior for the parameters is Gaussian with $\mu = 0.$ and $\sigma = 0.5$, transformed by a sigmoid function in the simulator to be positive and bounded. The resulting 4-dimensional posterior is highly correlated.

VAE: The decoder $g_\psi(x)$ of a Variational Autoencoder (VAE) was used as a generative model for handwritten digits (Kingma & Welling, 2014). The prior is a three-dimensional standard Gaussian. Images generated by the decoder were used as observed data (28×28 dimensional), with Gaussian observation noise with $\sigma = 0.05$:

$$p(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}; g_\psi(\boldsymbol{x}), \sigma^2 \mathbf{I}) \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \mathbf{I})$$

Due to the training procedure of the VAE, the posterior should be almost Gaussian.

Hodgkin Huxley: A neuroscience model that describes how action potentials in neurons are initiated and propagated. It is implemented based on Pospischil et al. (2008) and taken from Tejero-Cantero et al. (2020). We use a uniform prior constrained to biologically reasonable values. The observed data is the membrane voltage at 200 equally spaced time points, to which we added normally distributed noise with $\sigma = 0.1$, leading to a 7-dimensional posterior distribution.

SIR: A epidemiological model with two free parameters: The rate of recovery for infected individuals γ and the rate of new infections β for a population of $N = 5$. The solution satisfies the following differential equation

$$\frac{dS}{dt} = -\beta \frac{S \cdot I}{N} \quad \frac{dI}{dt} = \beta \frac{S \cdot I}{N} \gamma - \gamma I \quad \frac{dI}{dt} = \gamma I.$$

The observed data correspond to the number of infections at 50 equally spaced time points. We added log-normal observation noise with $\sigma = 0.2$. The prior was a Gaussian with $\sigma = 2.$ transformed by a sigmoid function, resulting in a complex 2-dimensional posterior.

Spatial SIR: An epidemiological model with a spatial dimension, similar to Hermans et al. (2022). The model is initialized with three infections at random locations on the grid. The infection then propagates to neighboring grid cells with probability β per time step. Infected people recover with probability γ at each time step. We observed a 30×30 grid of infected/non-infected regions, subject to a Beta noise model modeling the probability of being infected after a test. A logNormal prior with $\sigma = 0.5$ was used.

A1.3. Metrics

The expected coverage was evaluated as proposed in Cannon et al. (2022); Hermans et al. (2022). Given the $100(1-\alpha)\%$ highest posterior density region of the posterior estimate $\text{HPR}_{1-\alpha}$, we target to estimate $\mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})}\left[\mathbb{1}\{\boldsymbol{\theta} \in \text{HPR}_{1-\alpha}\}\right]$, where $\mathbb{1}$ is the indicator function. If the model is well-calibrated, the empirical coverage should match the nominal coverage $1 - \alpha$.

As in Cannon et al. (2022), we evaluate the coverage given (adversarially) perturbed data. Thus, the expected coverage becomes:

$$\mathbb{E}_{p(\tilde{\boldsymbol{x}})}\left[\mathbb{E}_{p(\boldsymbol{\theta} | \tilde{\boldsymbol{x}})}\left[\mathbb{1}\{\boldsymbol{\theta} \in \text{HPR}_{1-\alpha}\}\right]\right]$$

We use a Monte Carlo approximation to obtain $\text{HPR}_{1-\alpha}$ as described by Rozet et al. (2021); Deistler et al. (2022a) to efficiently estimated this quantity.

A2. Adversarial training and TRADES

We mainly compared against two well-known methods to defend against adversarial examples (additional defenses and results in Sec. A4.3).

Table 1. **Relative and absolute tolerance levels for each task.** The values of ϵ are multiples of the average standard deviation of prior predictives.

TASK	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.5$	$\epsilon = 1$	$\epsilon = 2$
GAUSSIAN LINEAR	0.11	0.21	0.32	0.53	1.05	2.13
SIR	0.03	0.06	0.09	0.15	0.3	0.6
LOTKA VOLTERRA	0.02	0.04	0.06	0.1	0.19	0.39
HODGKIN HUXLEY	1.83	3.65	5.48	9.13	18.26	36.54
VAE	0.02	0.05	0.07	0.12	0.25	0.5
SPATIAL SIR	0.02	0.04	0.07	0.11	0.22	0.45

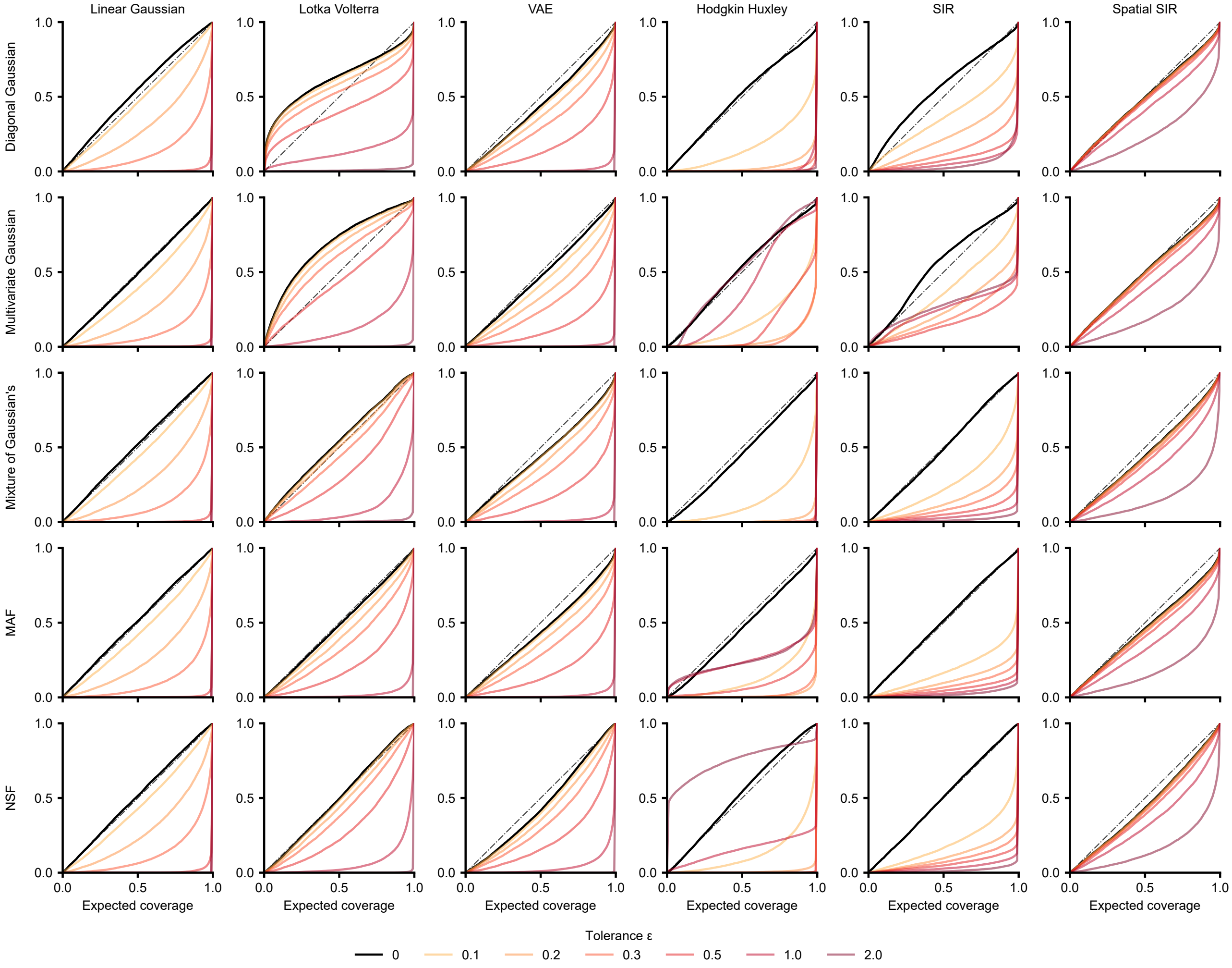

Figure A1. **Expected coverages for all density estimators.** Expected coverage metric for a specific density estimator trained with 10^5 simulations on standard NPE loss. The performance on well-specified data in black; colors indicate performance on adversarially perturbed data at certain tolerance levels.

A2.1. Adversarial training

Madry et al. (2018) proposed that the loss should be modified such that, instead of minimizing the negative log-likelihood (NLL) given the clean observation, one minimizes the NLL given the worst possible observation $\tilde{x}$ within an ϵ ball around the observation (i.e. the observation that has the highest NLL within the ϵ ball). Formally, this objective can be written as

$$\min_{\phi} \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})} \left[\max_{\tilde{\boldsymbol{x}} \in \{\tilde{\boldsymbol{x}} \mid ||\tilde{\boldsymbol{x}} - \boldsymbol{x}|| \leq \epsilon\}} - \log q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}) \right].$$

This scheme encourages the trained neural network to have a low NLL for any $\tilde{\boldsymbol{x}}$ within the ϵ-ball and thus encourages robustness to any adversarial perturbation.

In order to find the $\tilde{\boldsymbol{x}}$ with the highest NLL within the ϵ-ball, one commonly generates an adversarial example *during training* e.g., with projected gradient descent (Madry et al., 2018). Considering the adversarial perturbation as a distribution $p(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{\theta})$ gives:

$$\begin{aligned}\mathcal{L}(\phi) &= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})} \left[\mathbb{E}_{p(\tilde{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta})} \left[-\log q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}) \right] \right] \\ &= \mathbb{E}_{p(\tilde{\boldsymbol{x}},\boldsymbol{\theta})} \left[-\log q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}) \right]\end{aligned}$$

This reveals that by modifying the training scheme, q_ϕ no longer converges to the true posterior but to a different posterior distribution given by

$$\tilde{p}(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}) \propto \tilde{p}(\tilde{\boldsymbol{x}}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \int p(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{x}p(\boldsymbol{\theta})$$

i.e., the posterior distribution given the likelihood of observing the adversarially perturbed $\tilde{\boldsymbol{x}}$ given $\boldsymbol{\theta}$. Therefore, adversarial training can be interpreted as a regularization scheme where the data is perturbed by the adversarial perturbation (instead of a random perturbation as in Shu et al. (2018)).

In our experiments, we use an ℓ_2 projected gradient descent attack with 20 iterations during training. After initial experiments, we hand-picked $\epsilon = 0.1$ for the Gaussian linear task and $\epsilon = 1.0$ for the other benchmark tasks during training.

A2.2. TRADES

A second method for adversarial robustness was proposed by Zhang et al. (2019). Their proposed loss function balances the trade-off between performance on clean and adversarially perturbed observations, controlled through a hyperparameter β. The resulting surrogate loss adds a Kullback-Leibler divergence regularizer in the form of

$$\mathcal{L}(\phi) = \mathbb{E}_{p(\tilde{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{\theta})} \left[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) + \beta D_{KL}(q_\phi(\boldsymbol{\theta}|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}) \right].$$

This ensures that the posterior estimate is smooth (as measured by the KL divergence). However, this approach requires the ability to evaluate the KL divergence, which, for normalizing flows, can only be approximated through Monte Carlo techniques.

In general, the strength of regularization is determined by both the magnitude of the adversarial example, ϵ, and the hyperparameter β. Initial experiments showed that ϵ had a greater effect than β, so we only varied ϵ. To estimate the D_{KL} during training, we use a single Monte Carlo sample. We fixed $\beta = 0.1$ and hand-picked, based on initial experiments, $\epsilon = 0.1$ for the linear Gaussian and VAE, $\epsilon = 0.5$ for Hodgkin Huxley, Lotka Volterra, and Spatial SIR. The optimization to obtain $\tilde{\boldsymbol{x}}$ was run with an ℓ_2 projected gradient descent attack with 20 iterations.

A3. Tradeoff between posterior approximation and robustness

Adversarially robust models typically sacrifice accuracy on clean data in order to achieve robustness. The errors on clean and perturbed data have even been suggested to be fundamentally at odds (Zhang et al., 2019; Tsipras et al., 2019) and are subject to the required strength of adversarial robustness (Min et al., 2021), even in the infinite data limit.

Regularizing with the Fisher Information Matrix (FIM) creates a similar trade-off between accuracy on clean and perturbed data. The standard loss of neural posterior estimation minimizes

$$\mathcal{L}(\phi) = \mathbb{E}_{p(\boldsymbol{x})} \left[D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}|\boldsymbol{x})) \right] = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})} \left[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) \right]$$

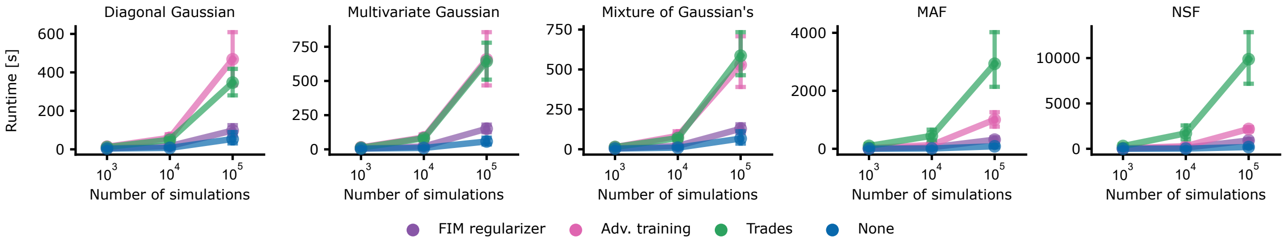

Figure A2. **Runtime.** Average runtime in seconds for each defense, calculated across all benchmark tasks and various hyperparameters per density estimator. For the MAF estimator with 5 Monte Carlo samples, the average timings for $100k$ training simulations are: 60 seconds for NPE, 250 seconds for FIM, 770 seconds for adv. training, and 2790 seconds for TRADES.

whereas the FIM regularizer minimizes

$$\Omega(\phi) = \mathbb{E}_{p(\boldsymbol{x},\tilde{\boldsymbol{x}})}\left[D_{KL}(q_\phi(\boldsymbol{\theta}|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}))\right].$$

$\Omega(\phi)$ is minimized globally if $q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) = q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}})$ for all $\boldsymbol{x}, \tilde{\boldsymbol{x}} \sim p(\boldsymbol{x})$. This suggests that the optimal $q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}})$ is independent of $\boldsymbol{x}$ and, thus, indeed at odds with approximating the posterior distribution given clean data.

The strength of this trade-off is determined by the value of the hyperparameter β, and the effect on the posterior fit is demonstrated in Fig. 5C, were we plot the trade-off between accuracy on clean data (evaluated as average log-likelihood) and the robustness to adversarial perturbations (measured as D_{KL} between clean and perturbed posteriors). The plotted values of β are Pareto-optimal solutions approximately solving the multi-objective optimization problem:

$$\min_{\beta}\left[\mathcal{L}(\phi_\beta^*), \Omega(\phi_\beta^*)\right] \text{ with } \phi_\beta^* = \arg\min_{\phi_\beta} \mathcal{L}(\phi_\beta) + \beta\mathbb{E}_{p(\boldsymbol{x})}\left[tr(\mathcal{I}_{\boldsymbol{x}})\right]$$

It is clear that a large value of β heavily regularizes Ω, pushing it towards zero, which results in the inference model ignoring the data and increasing $\mathcal{L}$. Thus, as β grows large, $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ approaches $p(\boldsymbol{\theta})$, i.e. the prior distribution, as this is the best estimate according to $\mathcal{L}$ which is independent of $\boldsymbol{x}$. This is in correspondence with approaches for robust generalized Bayesian inference with, e.g., α-posteriors (Grünwald & Ommen, 2017; Vovk, 1990; Medina et al., 2022).

However, for smaller values of β, there is a plateau where the accuracy of clean data is almost constant, but the levels of robustness vary significantly. This suggests that multiple inference models exist that have similar approximation errors but differ in their robustness to adversarial examples. The regularizer in this region can effectively induce robustness without sacrificing much accuracy.

It is worth noting that at a certain value of β, the approximation error increases significantly while the robustness decreases only gradually. As previously discussed in the main paper, β is closely related to the magnitude of the adversarial perturbation $\tilde{\boldsymbol{x}}$ i.e. the tolerance level ϵ. The *true* posterior might not be robust to such large-scale perturbations, making it an invalid solution subject to robustness constraints.

A4. Additional benchmark results

Here, we present additional results obtained on the benchmark tasks.

A4.1. Runtime

In Figure A2, we show the average runtime of the benchmark tasks. It can be observed that FIM regularization has a slightly higher cost compared to standard NPE, whereas TRADES is substantially more expensive across various density estimators. This especially holds for normalizing flows where the D_{KL} regularizer is estimated via Monte Carlo.

A4.2. Additional results on attacks

Which observations are particularly vulnerable to adversarial attacks? In our experiments, we generated a subset of parameters $\boldsymbol{\theta}$, along with well-specified data points $\boldsymbol{x}$, and adversarial examples found on these data points, $\tilde{\boldsymbol{x}}$. We

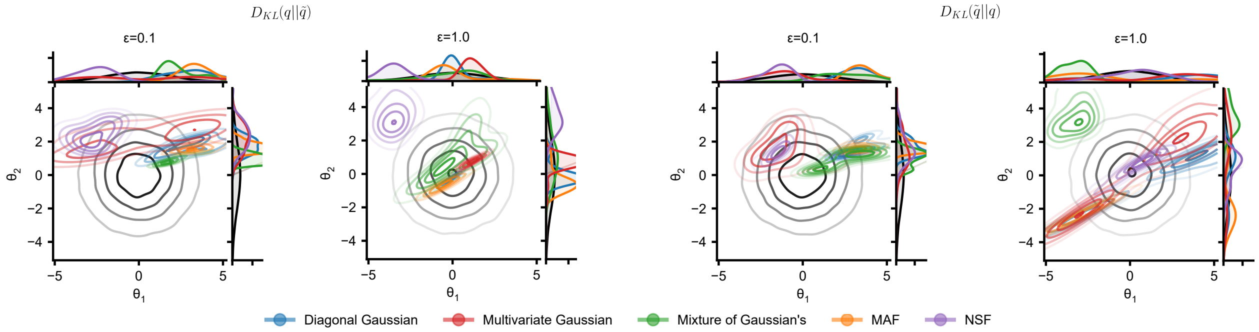

Figure A3. **Parameters which generated data susceptible to adversarial perturbations.** In black, we illustrate the prior over parameters $\boldsymbol{\theta}$ for the SIR task. For each density estimator trained with 10^5 simulations, we plot the distribution of $\boldsymbol{\theta}$'s on which we found the 50 "strongest" adversarial examples using the L2PGDAttack maximizing $D_{KL}(q||\tilde{q})$ (left) or $D_{KL}(\tilde{q}||q)$ (right).

next asked which observations are particularly vulnerable to adversarial attacks and what regions these correspond to in parameter space. We selected the 10% datapoints which had the highest D_{KL} between posterior estimates given clean and perturbed data and visualized the distribution of their corresponding ground truth parameters (Fig. A3). Attacks on different density estimators have higher efficacy on different sets of parameters, with some similarities (especially for similar density estimators such as Gaussian and Multivariate Gaussian). This indicates that the attacks not only leverage worst-case misspecification but also attack the particular neural network. Notably, parameters that generate data that is vulnerable to adversarial attacks are not necessarily found in the tails of the prior (where training data is scarce), but also in regions where many training data points are available. This observation could imply either vulnerable areas in the specific neural networks or/and susceptible regions within the generative model.

Are more complex density estimators less robust? We evaluated the D_{KL} (forward and backward) between posterior estimates given clean and perturbed data for several conditional density estimators. Our results show similar adversarial robustness across all tested density estimators (Fig. A4). We note that attacking simple models might appear to be more vulnerable because the adversarial objective can be computed in closed form, whereas complex models require Monte Carlo approximations.

We also computed the expected coverage for all density estimators (Fig. A1). Again, the expected coverages suggest a similar level of adversarial robustness across different conditional density estimators.

Does the adversarial objective matter? We evaluated whether using the forward vs the backward D_{KL} as the target for the adversarial attack influences the results. Despite minor differences, there is no clear advantage of divergence over the other. Adversaries with different objectives may find different adversarial examples that are more severe, as measured by their notion of "distance" between the distributions, as shown in Figure A4. As the KL divergence is locally symmetric, these differences are only noticeable for larger tolerance levels ϵ.

In addition, we evaluated an attack based on the Maximum-Mean discrepancy (MMD)

$$\delta^* = \arg\max_{\delta} MMD^2(q_\phi(\theta|\boldsymbol{x})||q_\phi(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{\delta})) \text{ s.t. } ||\delta|| \leq \epsilon.$$

We use the kernel MMD with an RBF kernel, estimated by a linear time Monte Carlo estimator as defined in Gretton et al. (2012) using ten samples. The MMD attack has a similar impact as D_{KL} attacks, but it is significantly weaker for some tasks, such as Lotka Volterra and SIR. One potential explanation for this could be an unsuitable kernel selection. Specifically, if the length scale of the RBF kernel is too small, it is well-known that the gradients tend to vanish. This issue can be particularly noticed in the SIR task, which plateaus for larger tolerance levels (in contrast to KL-based attacks, which explode). We note, however, that the MMD attack could also be applied to implicit density estimators (such as VAEs or GANs)

Finally, we evaluated an attack that minimizes the log-likelihood of the true parameters:

$$\delta^* = \arg\max_{\delta} -\log q_\phi(\boldsymbol{\theta}_o|\boldsymbol{x}_o+\boldsymbol{\delta}) \text{ s.t. } ||\delta|| \leq \epsilon.$$

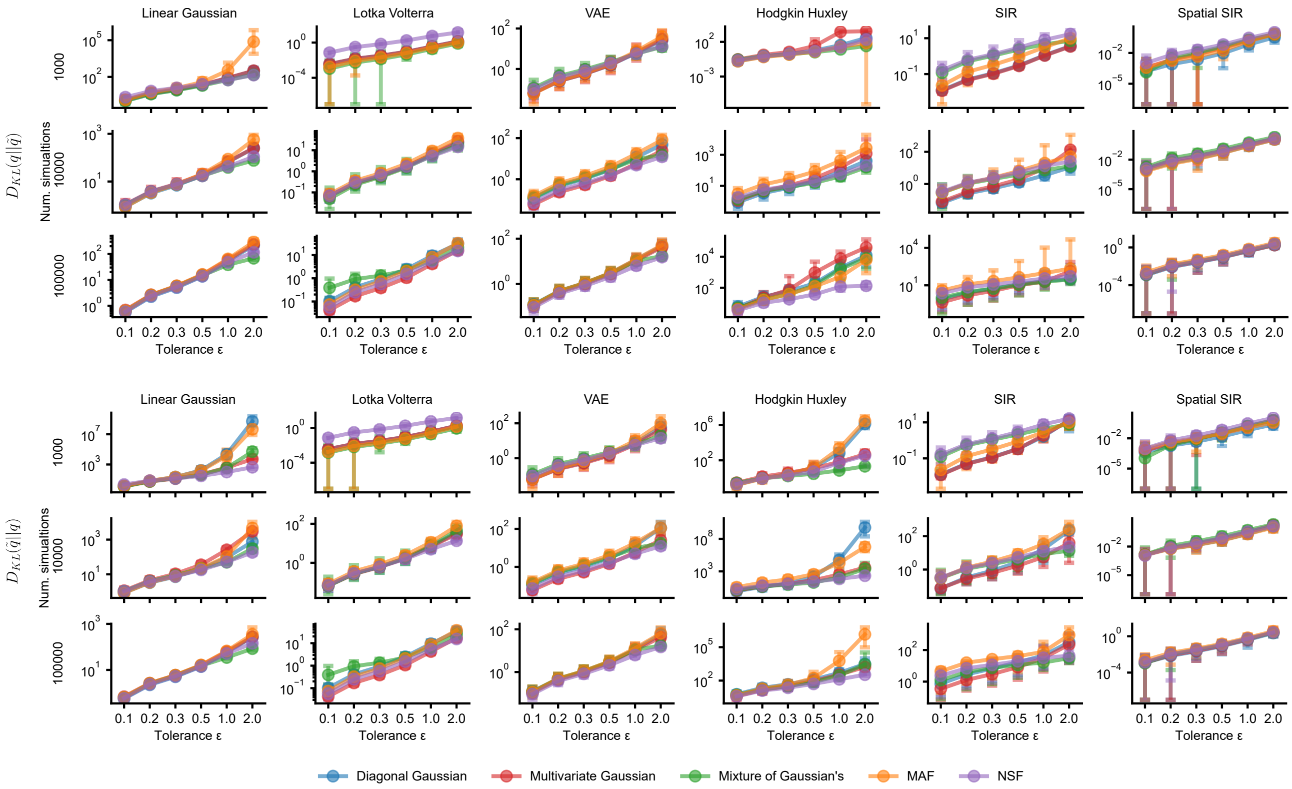

Figure A4. **Robustness for different density estimators.** Each density estimator is trained with NPE loss with a different number of simulations, attacked by an ℓ_2 projected gradient descent attack trying to maximize $D_{KL}(q||\tilde{q})$ (top) or $D_{KL}(\tilde{q}||q)$ (bottom).

Note that his attack requires access to one true parameter, which is only available for observations generated from the model and hence is not generally applicable. Furthermore, minimizing the likelihood of a single good parameter may not inevitably decrease the likelihood of all probable parameters. This attack strongly impacts the expected coverage since the attack objective is explicitly designed to avoid the true parameter and push it away from the region of the highest density (which is precisely the quantity measured by this metric).

A4.3. Additional results for the defenses

Robustness of ensembles and noise augmentation In addition to adversarial training and TRADES, we investigated two defense methods that were not originally developed as defenses against adversarial attacks: Ensembles and Noise Augmentation.

Ensembles do not use a single density estimator but K different ones. Assuming we trained all K density estimators to estimate the posterior distribution on different initialization, thus falling into different local minima. Then an Ensemble Posterior is typically defined as

$$q(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{k=1}^{K} \frac{1}{K} q_{\phi_k}(\boldsymbol{\theta}|x)$$

We built an ensemble of 10 masked autoregressive flows and evaluated its robustness to adversarial attacks on the benchmark tasks (Figure A7). The ensemble has a similar robustness as standard NPE.

Another defense, called 'Noise Augmentation', adds random perturbations to the data during training (in contrast to adversarial training, which uses adversarial perturbations). We use random noise uniformly distributed on the ℓ_2 ball with $\epsilon = 1.0$. Again, this defense only slightly (if at all) improved the robustness of NPE.

Overall, these results show that these defenses are not suitable to make amortized Bayesian inference robust to adversarial

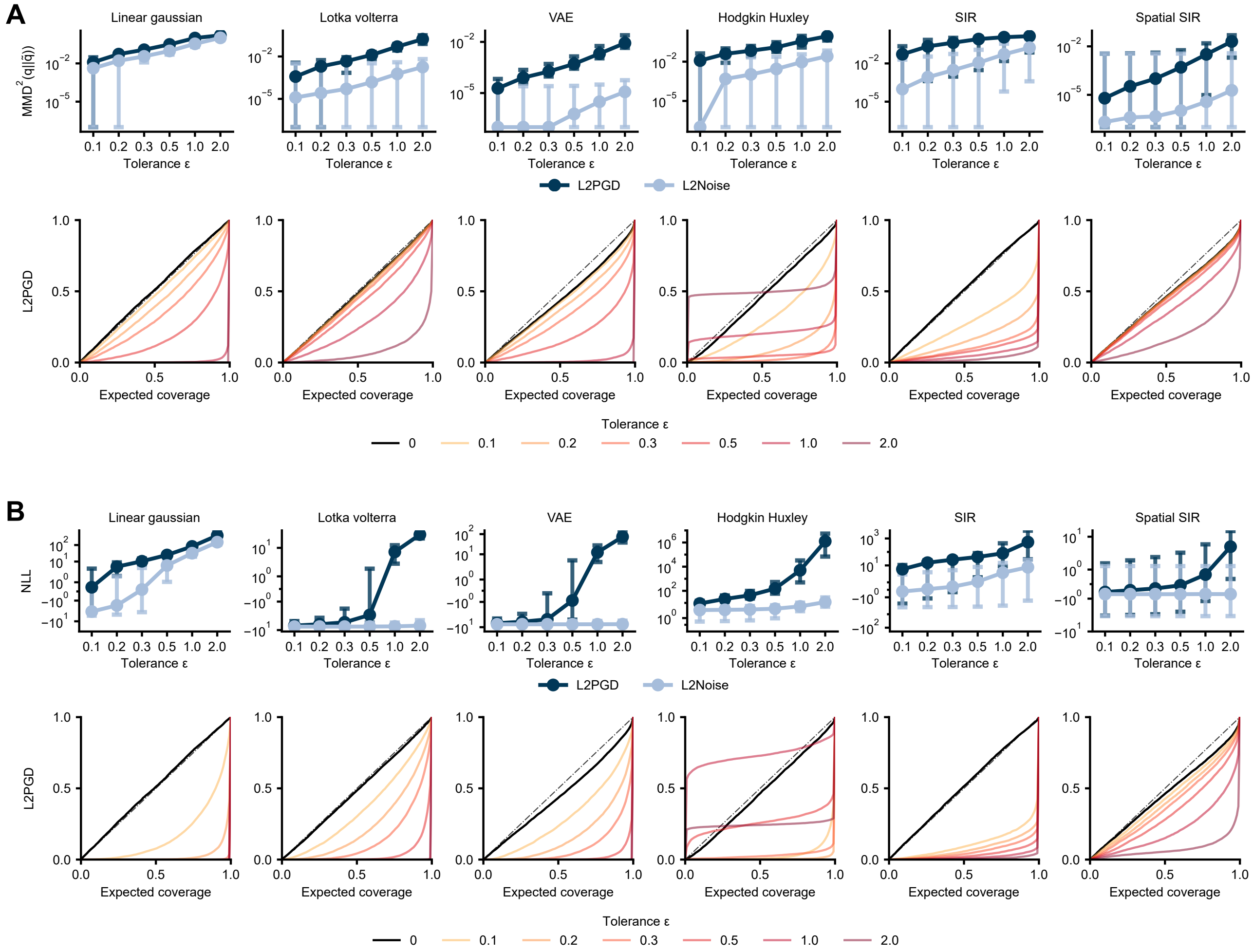

Figure A5. **Adversarial attack using different divergence.** Adversarial attacks using on each task using the MMD (**A**) and minimizing the true parameters log likelihood (**B**). On top, we show robustness $D_{KL}(q||\tilde{q})$ at the bottom; we show the expected coverage. The performance on well-specified data is shown in black; the colors indicate performance on adversarially perturbed data at certain tolerance levels.

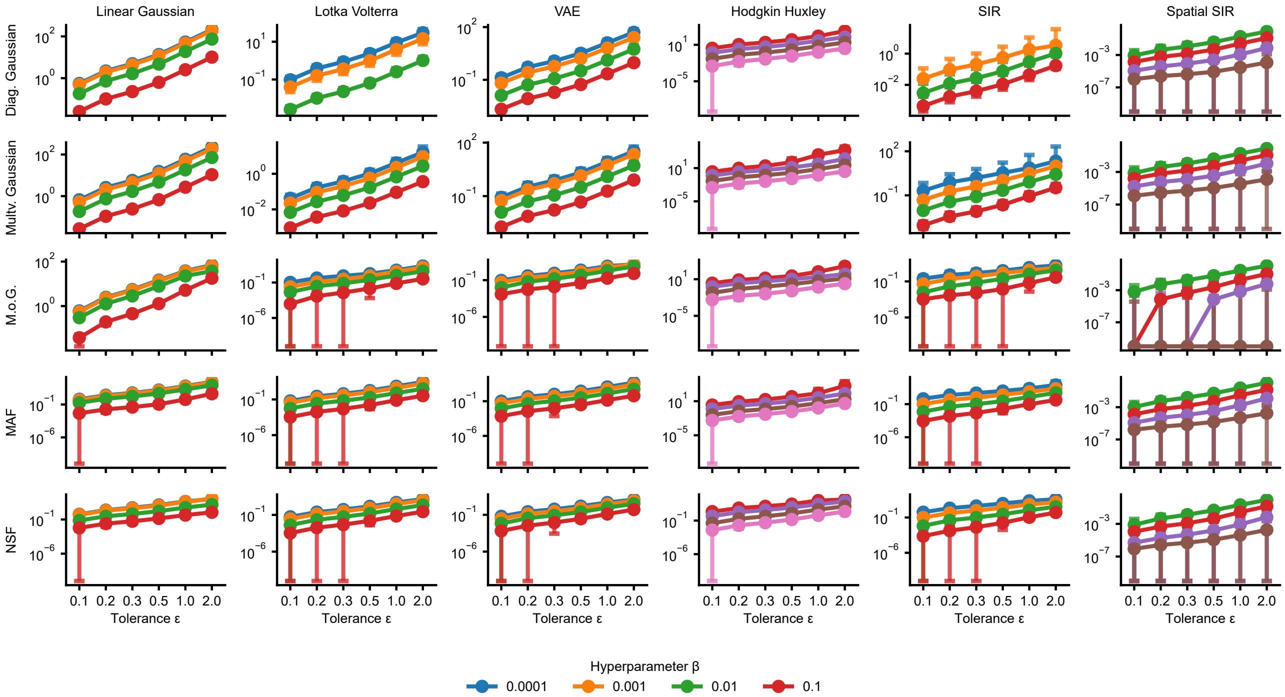

Figure A6. **Robustness of FIM.** Shows $D_{KL}(q||\tilde{q})$ for different density estimators (rows) and different values of regularization strength β (colors).

attacks.

Visualizing adversarial attacks with defenses. To visualize the effect on the inference of adversarial examples, we reproduced Figure 3 but with the robust inference models (with $\epsilon = 1.0$). FIM regularization is an effective defense while maintaining good accuracy on clean data, as evidenced by reasonable predictive distributions (Fig. A9). Notably, both adversarial training and TRADES do not work as well (Fig. A10, Fig. A11).

Please note that in these figures, we do not necessarily utilize the identical observation or posterior. Instead, we present visual representations of examples that share the same ”rank”. By sorting all adversarial perturbations based on their adversarial objective, we display the outcomes that correspond to the same index as selected for Figure 3. Pyloric network examples were additionally constrained to be biophysically realistic.

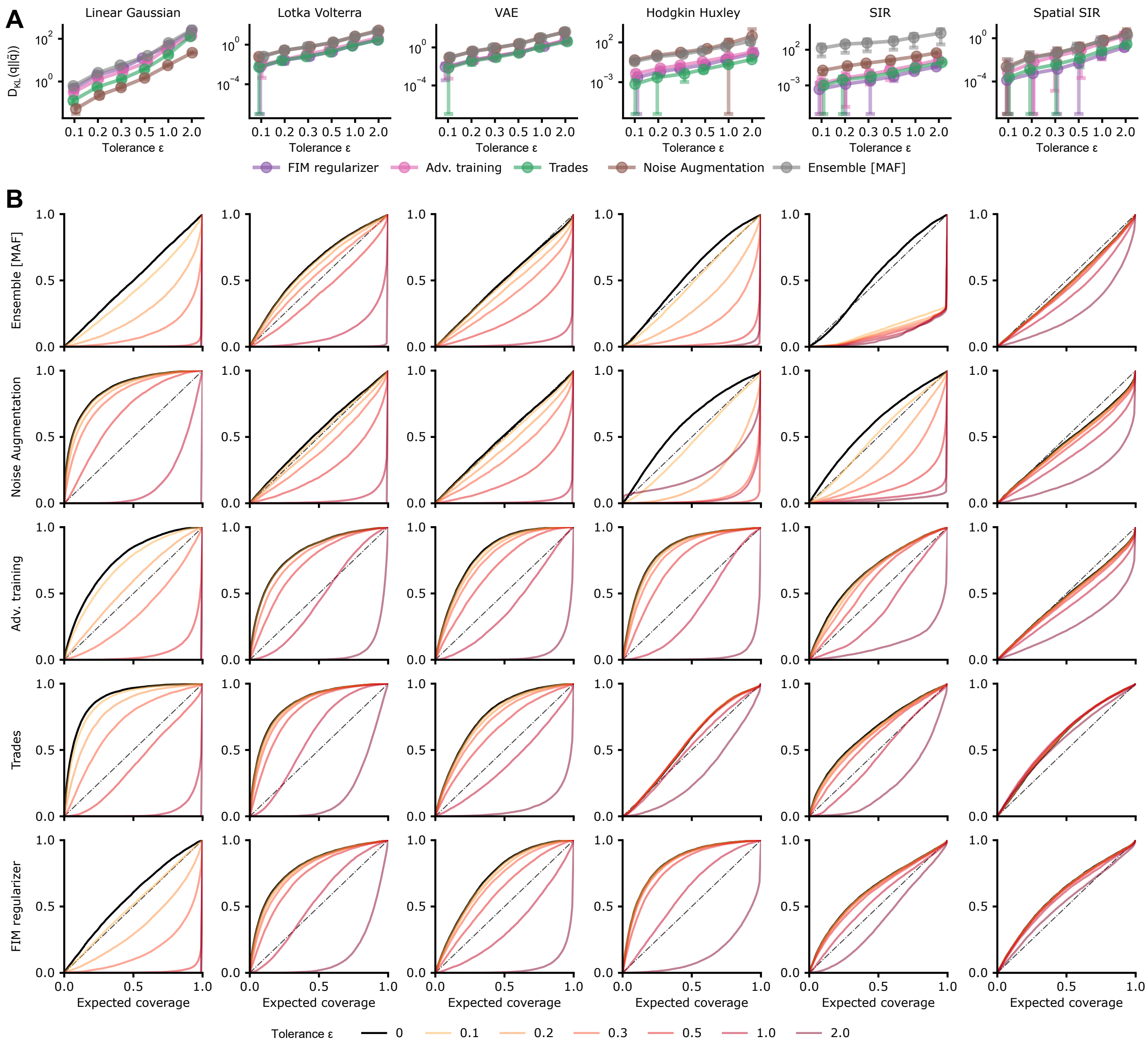

Figure A7. **Results for other defenses**. Robustness $D_{KL}(q||\tilde{q})$ for all defenses with selected hyperparameters (**A**). Expected coverage for all defenses with selected hyperparameters (**B**). The performance on well-specified data is shown in black; the colors indicate performance on adversarially perturbed data at certain tolerance levels.

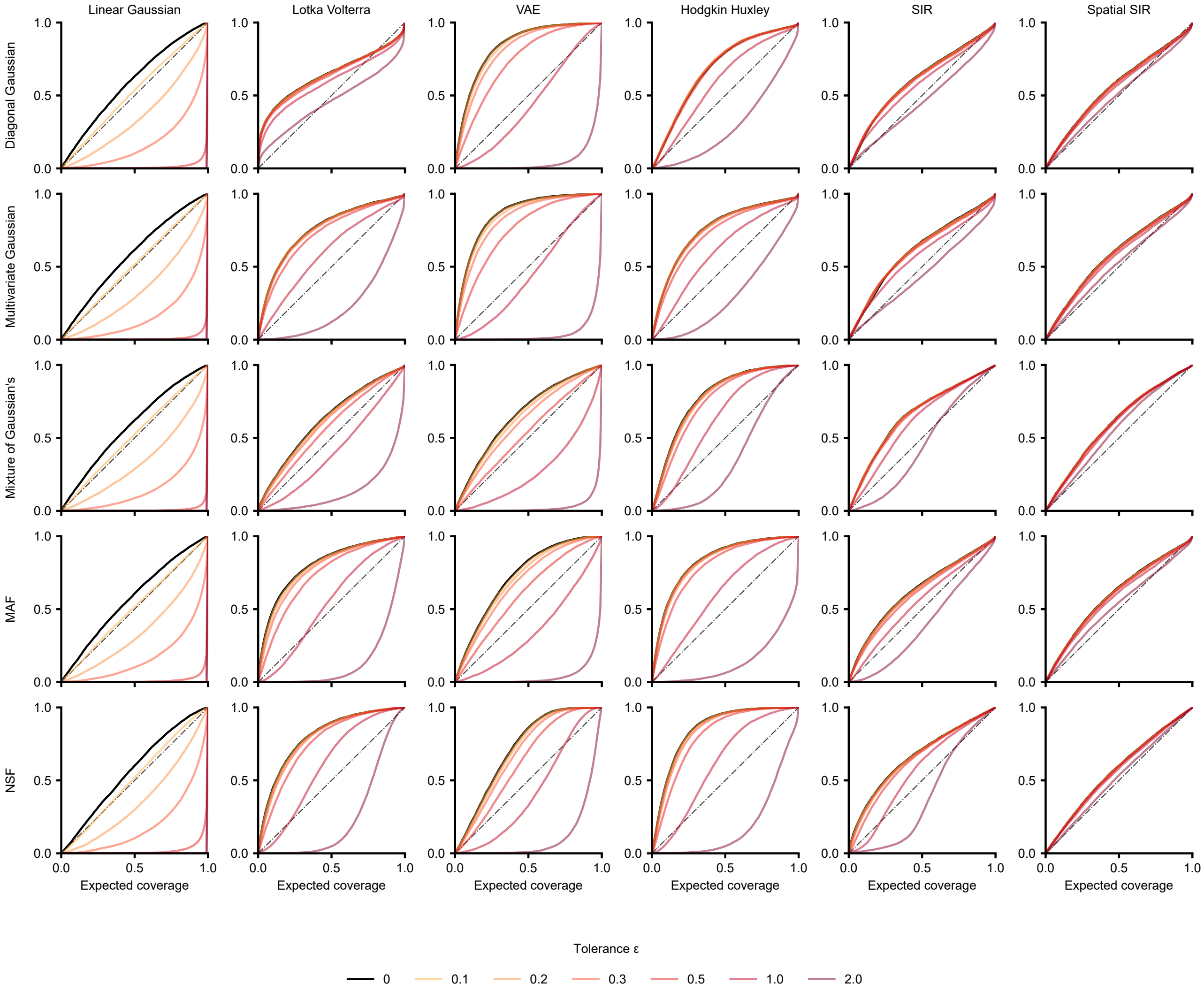

Figure A8. **Expected coverage for FIM regularization,** Each row shows the expected coverage for a specified density estimator regularized with trace of FIM, at selected hyperparameters. The performance on well-specified data is shown in black; the colors indicate performance on adversarially perturbed data at certain tolerance levels.

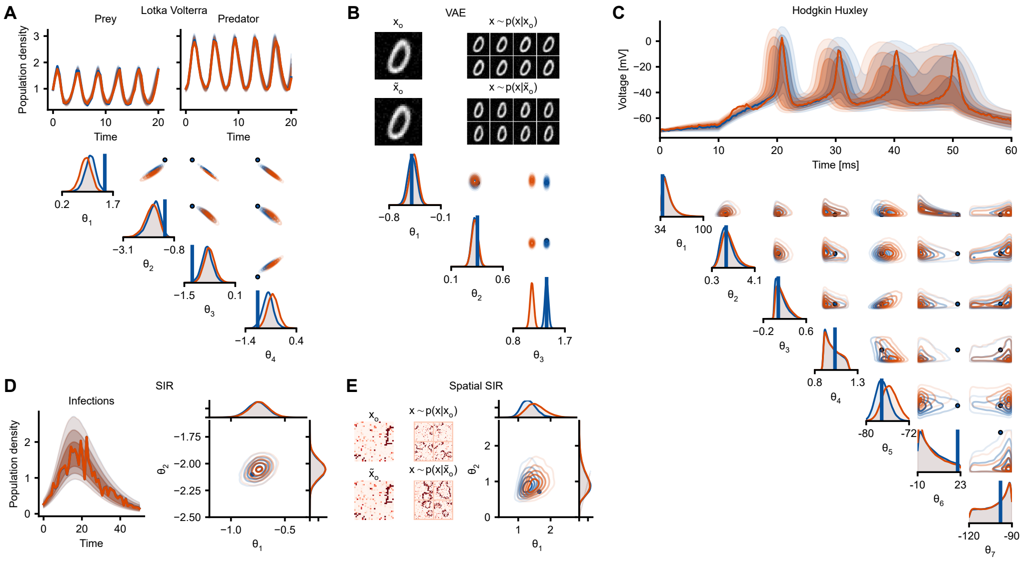

Figure A9. **Adversarial examples for each benchmark task when employing FIM regularization**. Each panel shows i) the original observation (blue line) and corresponding posterior predictive samples (blue shaded) ii) the adversarial example (orange line) and posterior predictive samples based on the perturbed posterior estimate, and iii) posterior distribution plots with the posterior estimate for the original (blue) and perturbed (orange) data, and the ground-truth parameters (black dot).

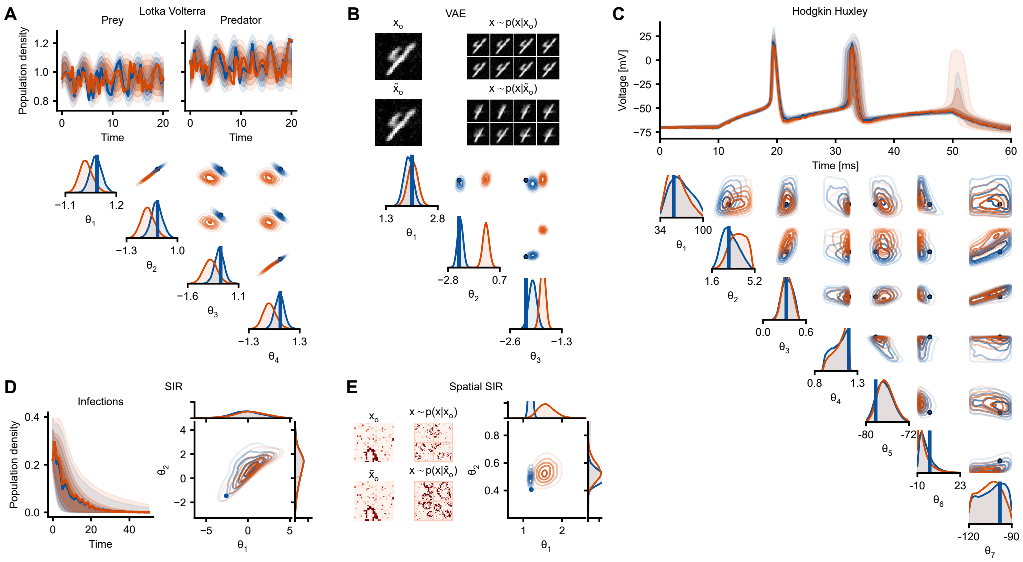

Figure A10. **Adversarial examples for each benchmark task when using adversarial training**. Each panel shows i) the original observation (blue line) and corresponding posterior predictive samples (blue shaded) ii) the adversarial example (orange line) and posterior predictive samples based on the perturbed posterior estimate, and iii) posterior distribution plots with the posterior estimate for the original (blue) and perturbed (orange) data, and the ground-truth parameters (black dot).

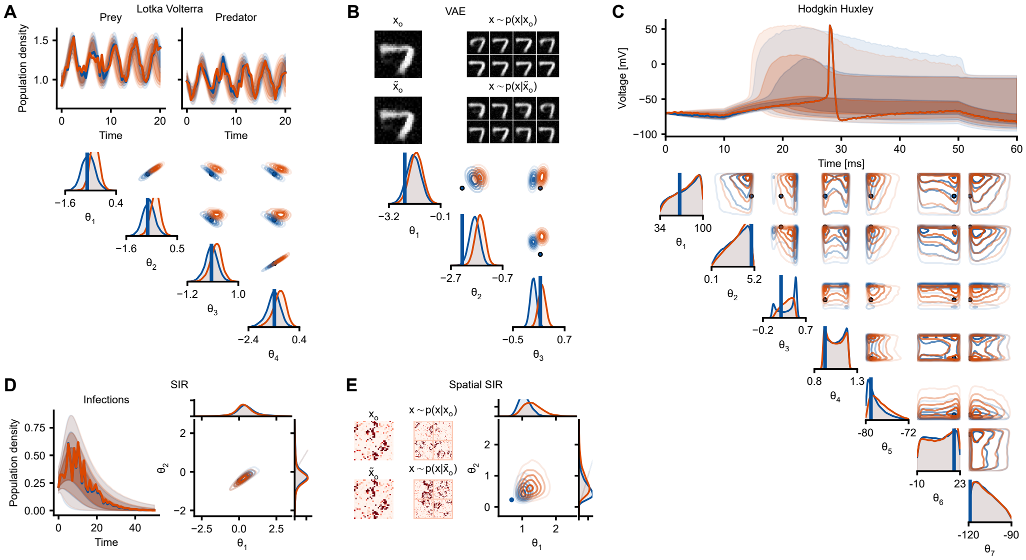

Figure A11. **Adversarial examples for each benchmark task when using TRADES**. Each panel shows i) the original observation (blue line) and corresponding posterior predictive samples (blue shaded) ii) the adversarial example (orange line) and posterior predictive samples based on the perturbed posterior estimate, and iii) posterior distribution plots with the posterior estimate for the original (blue) and perturbed (orange) data, and the ground-truth parameters (black dot).

A5. Optimal attacks for a linear Gaussian simulator

The only task in the benchmark with tractable ground-truth posterior for arbitrary $\boldsymbol{x}$ is the Gaussian linear task. Here we analyze this task in more detail. We will show that an attack that maximizes the D_{KL} between clean and perturbed posterior corresponds to the strongest eigenvector of the FIM. We then will compare the analytic robustness of the ground truth posterior to the empirical attack on NPE.

Assume the generative model is given by

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \text{ and } p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{b}, \boldsymbol{\Lambda}).$$

Then the posterior distribution is well-known and given by

$$p(\boldsymbol{\theta}|\boldsymbol{x}_o) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$$

$$\boldsymbol{\mu}_p(\boldsymbol{x}_o) = \boldsymbol{\mu}_0 + \boldsymbol{\Sigma}_0 \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\Sigma}_0\boldsymbol{A}^T + \boldsymbol{\Lambda})^{-1}(\boldsymbol{x}_o - (\boldsymbol{A}\boldsymbol{\mu}_0 + \boldsymbol{b})) = (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{A}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_o - \boldsymbol{b}) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0)$$
$$\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_0\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\Sigma}_0\boldsymbol{A}^T + \boldsymbol{\Lambda})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_0 = (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{A}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{A})^{-1}$$

As we will show below, for this simulator, the perturbation which maximizes the D_{KL} between clean and perturbed posterior exactly corresponds to the eigenvector of the Fisher-information matrix with the largest eigenvalue.

Analytical expression for the Kullback-Leibler divergence In this model, the Kullback-Leibler divergence $D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x}_o)||p(\boldsymbol{\theta}|\boldsymbol{x}_o + \boldsymbol{\delta}))$ between clean and perturbed posterior can be computed analytically.

We defined an adversarial example as a distorted observation given by $\tilde{\boldsymbol{x}}_o = \boldsymbol{x}_o + \boldsymbol{\delta}$. This will only affect the posterior mean, as the covariance matrix is independent of $\boldsymbol{x}_o$. The KL divergence between clean and perturbed posterior can be written as

$$\begin{aligned} D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x}_o)||p(\boldsymbol{\theta}|\boldsymbol{x}_o + \boldsymbol{\delta})) &= 0.5 \cdot \left(tr(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_p) - d + \log\left(\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_p|}\right)\right) \\ &\quad + 0.5(\boldsymbol{\mu}_p(\boldsymbol{x}_o) - \boldsymbol{\mu}_p(\boldsymbol{x}_o + \boldsymbol{\delta}))^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p(\boldsymbol{x}_o) - \boldsymbol{\mu}_p(\boldsymbol{x}_o + \boldsymbol{\delta})) \\ &= 0.5\left((\boldsymbol{\mu}_p(\boldsymbol{x}_o) - \boldsymbol{\mu}_p(\boldsymbol{x}_o + \boldsymbol{\delta}))^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p(\boldsymbol{x}_o) - \boldsymbol{\mu}_p(\boldsymbol{x}_o + \boldsymbol{\delta}))\right) \end{aligned}$$

and the difference between means can be written as

$$(\mu_p(\boldsymbol{x}_o) - \mu_p(\boldsymbol{x}_o + \boldsymbol{\delta})) = (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\delta} = \boldsymbol{\Sigma}_p \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\delta}.$$

Hence, we obtain

$$D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x}_o)||p(\boldsymbol{\theta}|\boldsymbol{x}_o + \boldsymbol{\delta})) = 0.5 \cdot \boldsymbol{\delta}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{A} \boldsymbol{\Sigma}_p \boldsymbol{\Sigma}_p^{-1} \boldsymbol{\Sigma}_p \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\delta} = 0.5 \cdot \boldsymbol{\delta}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{A} \boldsymbol{\Sigma}_p \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\delta}.$$

Analytical expression for the FIM Next, we derive a closed-form expression for the Fisher Information Matrix (FIM):

$$\mathcal{I}_{\boldsymbol{x}} = \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{x}} \log p(\boldsymbol{\theta}|\boldsymbol{x}) (\nabla_{\boldsymbol{x}} \log p(\boldsymbol{\theta}|\boldsymbol{x}))^T \right]$$

We can write:

$$\begin{aligned}
\nabla_{\boldsymbol{x}} \log p(\boldsymbol{\theta}|\boldsymbol{x}) &= -0.5 \nabla_{\boldsymbol{x}} (\mu_p(\boldsymbol{x}) - \boldsymbol{\theta})^T \boldsymbol{\Sigma}_p^{-1} (\mu_p(\boldsymbol{x}) - \boldsymbol{\theta}) \\
&= -(\nabla_{\boldsymbol{x}} \mu_p(\boldsymbol{x}))^T \boldsymbol{\Sigma}_p^{-1} (\mu_p(\boldsymbol{x}) - \boldsymbol{\theta}) \\
&= (\boldsymbol{\Sigma}_p \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1})^T \boldsymbol{\Sigma}_p^{-1} (\boldsymbol{\theta} - \mu_p(\boldsymbol{x})) \\
&= \boldsymbol{\Lambda}^{1-} \boldsymbol{A} (\boldsymbol{\theta} - \mu_p(\boldsymbol{x})) \\
&= \boldsymbol{\Lambda}^{-1} \boldsymbol{A} (\boldsymbol{\theta} - \mu_p(\boldsymbol{x}))
\end{aligned}$$

Hence the Fisher information matrix with respect to x is given by

$$\begin{aligned}
\mathcal{I}_{\boldsymbol{x}} &= \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{x}} \log p(\boldsymbol{\theta}|\boldsymbol{x}) (\nabla_{\boldsymbol{x}} \log p(\boldsymbol{\theta}|\boldsymbol{x}))^T \right] \\
&= \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})} [\boldsymbol{\Lambda}^{-1} \boldsymbol{A} (\boldsymbol{\theta} - \mu_p(\boldsymbol{x})) (\boldsymbol{\theta} - \mu_p(\boldsymbol{x}))^T \boldsymbol{A}^T \Lambda^{-1}] \\
&= \boldsymbol{\Lambda}^{-1} \boldsymbol{A} \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})} [(\boldsymbol{\theta} - \mu_p(\boldsymbol{x})) (\boldsymbol{\theta} - \mu_p(\boldsymbol{x}))^T] \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1} \\
&= \boldsymbol{\Lambda}^{-1} \boldsymbol{A} \boldsymbol{\Sigma}_p \boldsymbol{A}^T \boldsymbol{\Lambda}^{-1}
\end{aligned}$$

and thus equivalently

$$D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x}_o)||p(\boldsymbol{\theta}|\boldsymbol{x}_o + \boldsymbol{\delta})) = 0.5 \cdot \boldsymbol{\delta}^T \mathcal{I}_{\boldsymbol{x}} \boldsymbol{\delta}$$

This demonstrates that the Kullback-Leibler divergence between clean and perturbed posterior directly corresponds to the Fisher information matrix in the linear Gaussian simulator.

Optimal attack on the linear Gaussian simulator When maximizing the $D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x}_o)||p(\boldsymbol{\theta}|\boldsymbol{x}_o + \boldsymbol{\delta}))$, the adversary, thus, tries to solve the following problem

$$\boldsymbol{\delta}^* = \max_{\boldsymbol{\delta}:||\boldsymbol{\delta}||_2 \leq \epsilon} 0.5 \boldsymbol{\delta}^T \mathcal{I}_{\boldsymbol{x}} \boldsymbol{\delta}.$$

By Reyleight's theorem, this is solved by $\boldsymbol{\delta}^* = \epsilon \boldsymbol{v}_{max}$ where $\boldsymbol{v}_{max}$ is the eigenvector with the largest eigenvalue of $\mathcal{I}_{\boldsymbol{x}}$ and thus

$$D_{KL}(p(\boldsymbol{\theta}|\boldsymbol{x}_o)||p(\boldsymbol{\theta}|\boldsymbol{x}_o + \boldsymbol{\delta})) \leq 0.5 \lambda_{max} \epsilon^2.$$

Here λ_{max} is the maximum eigenvalue of $\mathcal{I}_{\boldsymbol{x}}$. Thus any empirical attack on the ground truth posterior would be bounded by this quantity.

An attack on an inference model that successfully identifies the map to the ground truth posterior should be consistent with this result. Figure A12 illustrates the ground-truth robustness to ℓ_2 perturbations (black) and the empirical robustness obtained through attacking NPE inference models attempting to solve this task. As the number of simulations increases, the NPE model can more accurately capture the true posterior mapping, resulting in attacks that are bounded by this quantity. This is evident in the figure, where attacks on most models are close to this bound. However, if the model is trained on too few simulations, the results may differ greatly as an incorrect mapping to the posterior is learned, making it more susceptible to adversarial attacks. Interestingly, given insufficient training data, all density estimators tend to be more, and not less, brittle to adversarial perturbations. The attacks are weaker on Mixture models and neural spline flows, which could result from either the attack not being strong enough or the model being too smooth.

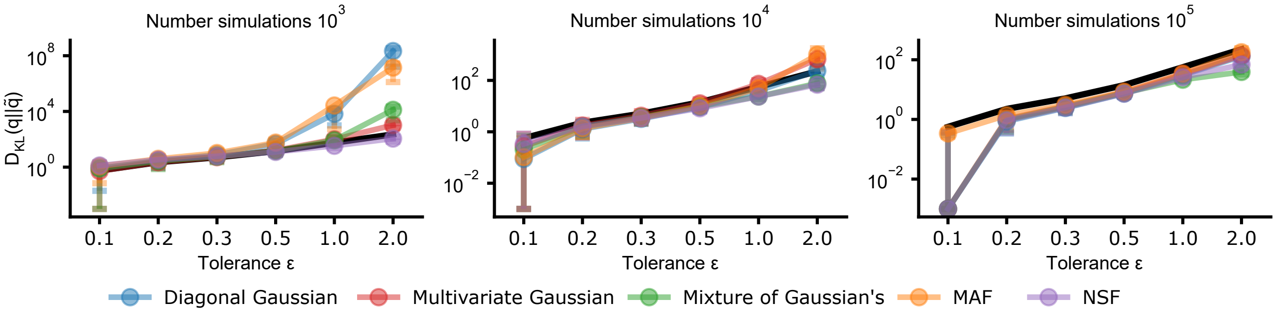

Figure A12. **Optimal attack on Gaussian linear posterior.** Gaussian linear attacks example on multiple density estimators. We show the ”robustness” of the true posterior as a black line.

A6. Analytical expression for the FIM regularization in generalized linear models

We analyze the solution on a generalized linear Gaussian density estimator to investigate the bias introduced by FIM regularization. We derive the analytical solutions for the optimal parameters identified by NPE and NPE with FIM regularization and discuss their differences.

Consider a generalized linear Gaussian inference network

$$q_{\boldsymbol{W},\boldsymbol{\Sigma}}(\boldsymbol{\theta}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{W}\phi(\boldsymbol{x}), \boldsymbol{\Sigma})$$

here ϕ is a, possibly nonlinear, *feature mapping* $\phi : \mathbb{R}^{d_{\boldsymbol{x}}} \to \mathbb{R}^{d_\phi}$. The only learnable parameter is the weight matrix $W \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times d_\phi}$ and covariance matrix $\Sigma \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times d_{\boldsymbol{\theta}}}$.

In this case, the NPE loss using N simulations $(\boldsymbol{x}_i, \boldsymbol{\theta}_i)$, can be written as

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{W}, \boldsymbol{\Sigma}) = \frac{1}{2} tr\left((\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta}))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta}) \right) + \frac{N}{2} \log\det(\boldsymbol{\Sigma})$$

here $\boldsymbol{X} \in \mathbb{R}^{d_x \times N}$ denotes all data points represented as columns of a matrix (equivalently $\boldsymbol{\Theta}$).

Below, we compute analytical expressions for the optimal parameters $\boldsymbol{W}, \boldsymbol{\Sigma}$ for (1) NPE and (2) for NPE with FIM-regularization. This allows us to quantify the bias introduced by FIM-regularization in a Gaussian GLM.

Convergence of NPE For NPE (without regularization), we can compute the optimal parameters in closed-form:

$$\nabla_{\boldsymbol{W}} \mathcal{L} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{W}\phi(\boldsymbol{X})^T - \boldsymbol{\Theta})\phi(\boldsymbol{X})^T \overset{!}{=} 0$$
$$\iff \hat{\boldsymbol{W}} = \boldsymbol{\Theta}\phi(\boldsymbol{X})^T \left(\phi(\boldsymbol{X})\phi(\boldsymbol{X})^T \right)^{-1}$$

Which is a generalized linear least square regression estimator. Equivalently we can obtain an estimator for the covariance matrix:

$$\nabla_{\boldsymbol{\Sigma}} \mathcal{L} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}(\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta})(\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta})^T\boldsymbol{\Sigma}^{-1} + \frac{N}{2}\boldsymbol{\Sigma}^{-1} \overset{!}{=} 0$$
$$\iff N\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta})(\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta})^T\boldsymbol{\Sigma}^{-1}$$
$$\iff \hat{\boldsymbol{\Sigma}} = \frac{1}{N}(\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta})(\boldsymbol{W}\phi(\boldsymbol{X}) - \boldsymbol{\Theta})^T.$$

As $\hat{\boldsymbol{W}}$ is estimated independently of $\boldsymbol{\Sigma}$, we can plug in $\boldsymbol{W} = \hat{\boldsymbol{W}}$ to globally minimize the loss.

Convergence of NPE with FIM regularization In this model, we can also compute the FIM regularized solution in closed-form. The FIM is given by

$$\mathcal{I}_{\boldsymbol{x}} = J_\phi(\boldsymbol{x})^T \boldsymbol{W}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{W} J_\phi(\boldsymbol{x}).$$

Here $J_\phi(\boldsymbol{x})$ denotes the Jacobian matrix of ϕ at $\boldsymbol{x}$. Hence the FIM regularized model minimizes the loss

$$\mathcal{L}_{FIM}(\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{W}, \boldsymbol{\Sigma}, \beta) = \mathcal{L}(\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{W}, \boldsymbol{\Sigma}) + \frac{\beta}{N} \sum_{i=1}^{N} tr(\mathcal{I}_{\boldsymbol{x}_i}).$$

To avoid clutter in notation, let $\Omega(\boldsymbol{X}) = \frac{1}{N} \sum_{i=1}^{n} J_\phi(\boldsymbol{x}_i) J_\phi(\boldsymbol{x}_i)^T$. Then we can write a solution that minimizes this loss as

$$\begin{aligned} \nabla_{\boldsymbol{W}} \mathcal{L}_{FIM} =& \nabla_{\boldsymbol{W}} \mathcal{L} + 2\beta \boldsymbol{\Sigma}^{-1} \boldsymbol{W} \Omega(\boldsymbol{X}) \overset{!}{=} 0 \\ \iff \hat{\boldsymbol{W}}_{FIM} =& \boldsymbol{\Theta} \phi(\boldsymbol{X})^T \left(\phi(\boldsymbol{X}) \phi(\boldsymbol{X})^T + 2\beta \Omega(\boldsymbol{X}) \right)^{-1} \end{aligned}$$

and

$$\begin{aligned} \nabla_{\boldsymbol{\Sigma}} \mathcal{L}_{FIM} =& \nabla_{\boldsymbol{\Sigma}} \mathcal{L} - \beta \boldsymbol{\Sigma}^{-1} \boldsymbol{W} \Omega(\boldsymbol{X}) \boldsymbol{W}^T \boldsymbol{\Sigma}^{-1} \overset{!}{=} 0 \\ \iff \hat{\boldsymbol{\Sigma}}_{FIM} =& \hat{\boldsymbol{\Sigma}} + \frac{4\beta}{N} \boldsymbol{W} \Omega(\boldsymbol{X}) \boldsymbol{W}^T. \end{aligned}$$

Again we can plug in $\hat{\boldsymbol{W}}_{FIM}$ to globally minimize the loss.

Bias introduced by FIM-regularization The solution for W corresponds to a Tikhonov regularized least squares solution, which simplifies to ridge regression in the linear case (Golub et al., 1999). Recent research has shown a connection between adversarial training and ridge regression in the linear-least squares setting. (Ribeiro et al., 2022). The regularization approach based on Fisher Information Matrix (FIM) incorporates a bias term through the average Jacobian outer product, which is known to recover those directions that are most relevant to predicting the output (Trivedi & Wang, 2020). Thus, the regularization strength is directed towards the directions to which the feature mapping ϕ is most sensitive. This bias increases monotonically with the regularization parameter, leading to an asymptotically to smooth mean function.

Additionally, FIM regularization overestimates the covariance matrix for a finite number of data points. Interestingly, as $N \to \infty$ the additive bias on the covariance matrix vanishes. This is in agreement with our empirical results of 'conservative' posterior approximations.

A7. FIM approximations

We propose an efficient approximation method to scale FIM-based approaches to complex density estimators. Here we investigate the effect of the approximation compared to an exact method. In order to compare our method to a closed-from FIM, we use a simple Gaussian density estimator of the form

$$q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), diag(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))).$$

Here $\boldsymbol{\mu} : \mathbb{R}^{d_{\boldsymbol{x}}} \to \mathbb{R}^{d_{\boldsymbol{x}}}$ is the mean function parameterized as neural network and $\boldsymbol{\sigma} : \mathbb{R}^{d_{\boldsymbol{x}}} \to \mathbb{R}^{d_{\boldsymbol{x}}}$ the standard deviation, which is transformed to a diagonal covariance matrix. Let us denote the distributional parameters as

$$\boldsymbol{\eta} = \begin{pmatrix} \boldsymbol{\mu}_\phi(\boldsymbol{x}) \\ \boldsymbol{\sigma}_\phi(\boldsymbol{x}) \end{pmatrix}$$

and let J be the Jacobian matrix defined by $J_{ij} = \frac{\partial \boldsymbol{\eta}_i}{\partial x_i}$. Then we can write $\mathcal{I}_{\boldsymbol{x}} = J^T \mathcal{I}_{\boldsymbol{\eta}} J$. Notice that $\mathcal{I}_{\boldsymbol{\eta}}$ is a Gaussian FIM with respect to the parameter $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ which is given by

$$\mathcal{I}_{\boldsymbol{\eta}} = -\mathbb{E}_{q_{\boldsymbol{\eta}}} \left[\nabla^2_{\boldsymbol{\eta}} \log q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \right] = \begin{pmatrix} diag(\frac{1}{\boldsymbol{\sigma}^2}) & 0 \\ 0 & diag(\frac{2}{\boldsymbol{\sigma}^2}) \end{pmatrix}.$$

The Jacobian matrix can be computed via autograd and thus $\mathcal{I}_{\boldsymbol{x}}$ can be computed for any given $\boldsymbol{x}$.

To test how well our approximations work, we tested the following regularizers:

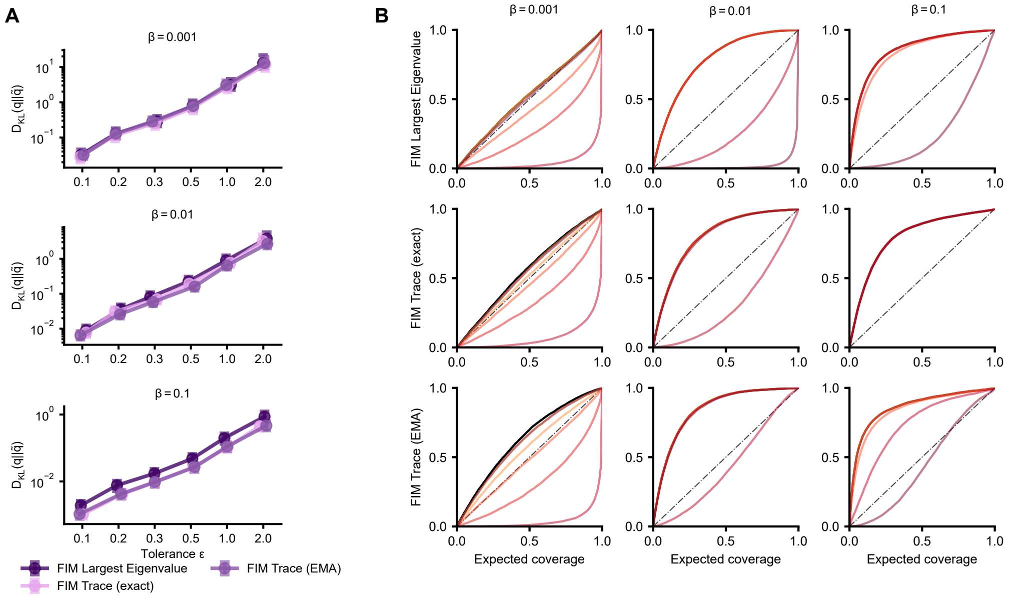

Figure A13. **FIM approximation:** Columns show the robustness and expected coverage for different choices of regularization strength β. Top row shows robustness $D_{KL}(q||\tilde{q})$. The bottom rows show the expected coverages of the FIM Largest Eigenvalue, FIM Trace (exact), and FIM Trace (EMA) regularizers.

- FIM Largest Eigenvalue: Using the regularize $\Omega(\boldsymbol{x}) = \beta \cdot \lambda_{\max}(\mathcal{I}_{\boldsymbol{x}})$ with the exact FIM.
- FIM Trace: This uses $\Omega(\boldsymbol{x}) = \beta \cdot tr(\mathcal{I}_{\boldsymbol{x}})$ also with the exact FIM.
- FIM Trace [EMA]: This uses $\Omega(\boldsymbol{x}) = \beta \cdot tr(\hat{\mathcal{I}}_{\boldsymbol{x}})$ estimated as described in the main paper.

The results on the VAE task are shown in Figure A13 using three different choices $\beta = 0.001, 0.01, 0.1$. Notably, for small regularization strengths, all techniques work similarly. Both trace-based regularizers decrease $D_{KL}(q||\tilde{q})$ more strongly, as expected by the fact that they are upper bounds of the largest eigenvalue. Notably, train time increased to five hours using the exact eigenvalue, compared to under two minutes using our approach.

Overall, these results demonstrate that our approximations to the largest eigenvalue of the FIM incur only a small cost in adversarial robustness.

A8. Comparission to MCMC based posteriors

We emphasize metrics that can be efficiently computed without requiring direct access to the true posterior. This choice is justified because numerous tasks lack a tractable or ”expensive to evaluate” likelihood. Consequently, calculating the posterior individually for thousands of observations would be computationally expensive. However, it is worth noting that we have a likelihood for some of the benchmark tasks and can perform non-amortized posterior computation with a manageable computational burden.

For the Gaussian Linear task, we have the advantage of an analytic solution. This simplifies the computation of the posterior, as we can directly derive the necessary quantities without relying on approximation methods or sampling techniques. In the case of the SIR, VAE, and Lotka Volterra tasks, we have tractable likelihood functions. This enables us to estimate the posterior distribution using likelihood-based inference methods, such as MCMC. We use a two-stage procedure to get good posterior approximations. First, we run one hundred parallel MCMC chains initialized from the prior. For the SIR and Lotka Volterra, we used an adaptive Gaussian Metropolis Hasting MCMC method (Andrieu & Thoms, 2008). For the VAE task, we used a Slice sampler (Neal, 2003). In the second stage, these results were used to train an unconditional flow-based density estimator, which then performs Independent Metropolis Hasting MCMC to obtain the final samples (Holden et al., 2009).

We present an illustration of several approximated (adversarial) posteriors alongside their corresponding ground truth obtained through Markov Chain Monte Carlo (MCMC) methods (Figure A14). As discussed in the main paper, even minor perturbations can lead to significant misspecification of the SIR model. We observe substantial changes in the true posterior distribution due to adversarial perturbations (A14A). Notably, the standard NPE method appears to be highly susceptible to this scenario, often producing predictions that seem arbitrary. In contrast, the FIM regularized inference networks exhibit a similar trend as the true posterior; however, they tend to be underconfident in their predictions as expected.

Figure A14B illustrates the behavior of the variational autoencoder (VAE) task in the presence of adversarial perturbations. Interestingly, the ground-truth posterior distribution remains remarkably unaffected by these perturbations, exhibiting a high degree of invariance. Yet, the inference network, responsible for approximating the posterior, proves to be susceptible to being deceived by these adversarial perturbations. This effect is strongly reduced for FIM regularization. We observe this similarly on the Lotka Volterra task.

We quantified this difference on a randomly selected set of 100 pairs $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ by computing $MMD^2(q,p) = MMD^2(q_\phi(\boldsymbol{\theta}|\boldsymbol{x}), p_{MCMC}(\boldsymbol{\theta}|\boldsymbol{x}))$ and $MMD^2(q,p) = MMD^2(q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}), p_{MCMC}(\boldsymbol{\theta}|\tilde{\boldsymbol{x}}))$ (Figure A15). Consistent with expectations, the MMD value for well-specified data demonstrates a relatively good agreement between the true posterior and the inference network's approximation across different tasks. However, when confronted with adversarially perturbed data, the MMD value increases significantly. This indicates that the inference network struggles to accurately capture the underlying posterior distribution in the presence of adversarial perturbations. These findings highlight the limitations of the inference network and its susceptibility to such adversarial perturbations. FIM regularization can mitigate this effect, effectively reducing the impact of adversarial perturbations on the inference network's performance. However, this improvement comes at the expense of decreased approximation quality on well-specified data.

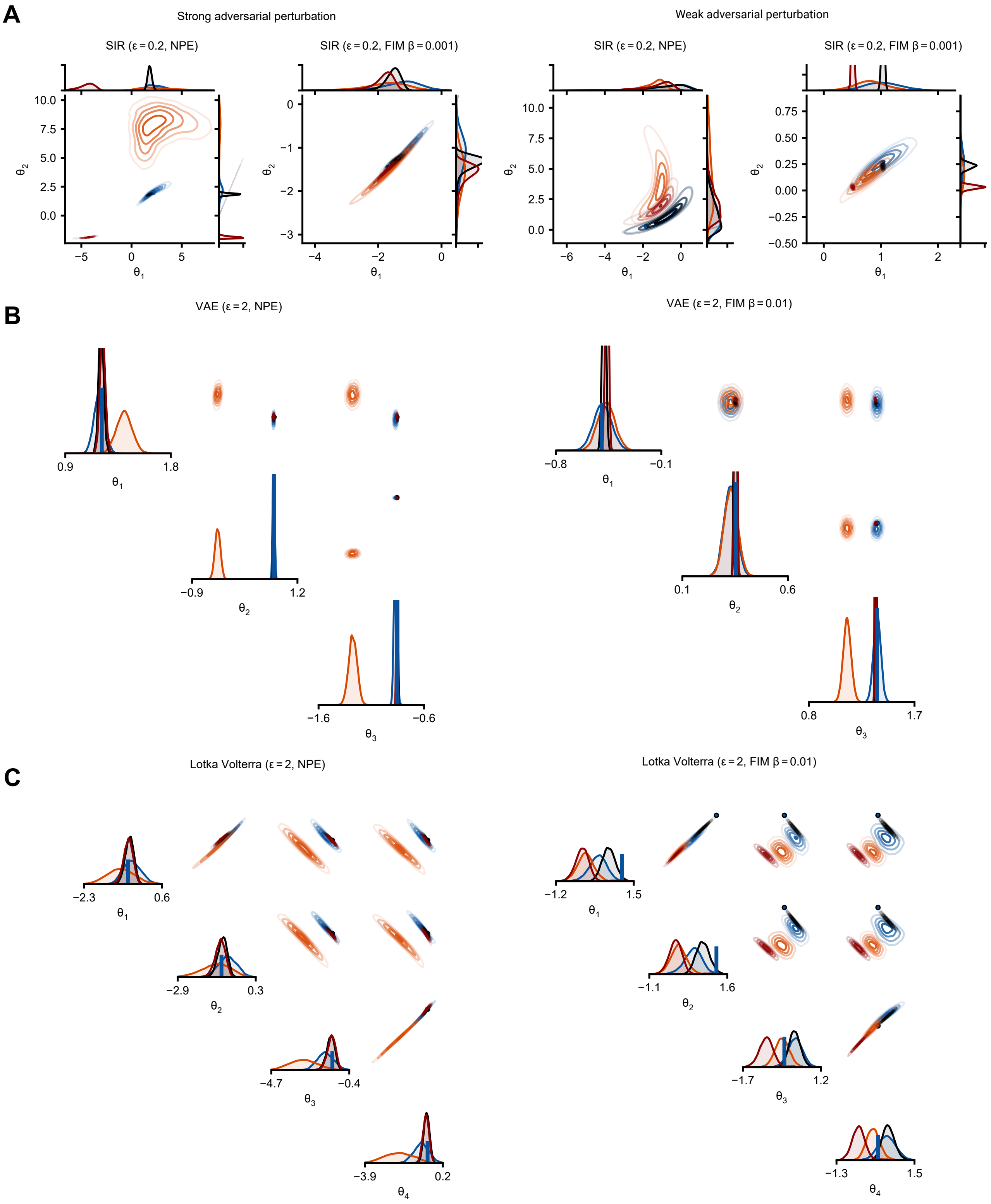

Figure A14. **Approximate posterior distributions plotted against MCMC-based posterior estimate.** This figure compares approximate amortized posterior estimates and the posterior obtained through MCMC sampling. In blue and orange, we can see the approximate amortized posterior estimate on the clean observations $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ and a perturbed observation $q_\phi(\boldsymbol{\theta}|\tilde{\boldsymbol{x}})$. In black, we plot the true posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ and in red, the true adversarial posterior $p(\boldsymbol{\theta}|\tilde{x})$ (both estimated via MCMC). We present this comparison for the SIR task, showcasing both a strong adversarial perturbation and weaker perturbation (as measured by $D_{KL}(q||\tilde{q})$). For the VAE and Lotka Volterra tasks, we only display the results for strong perturbations (denoted as **B** and **C**, respectively).

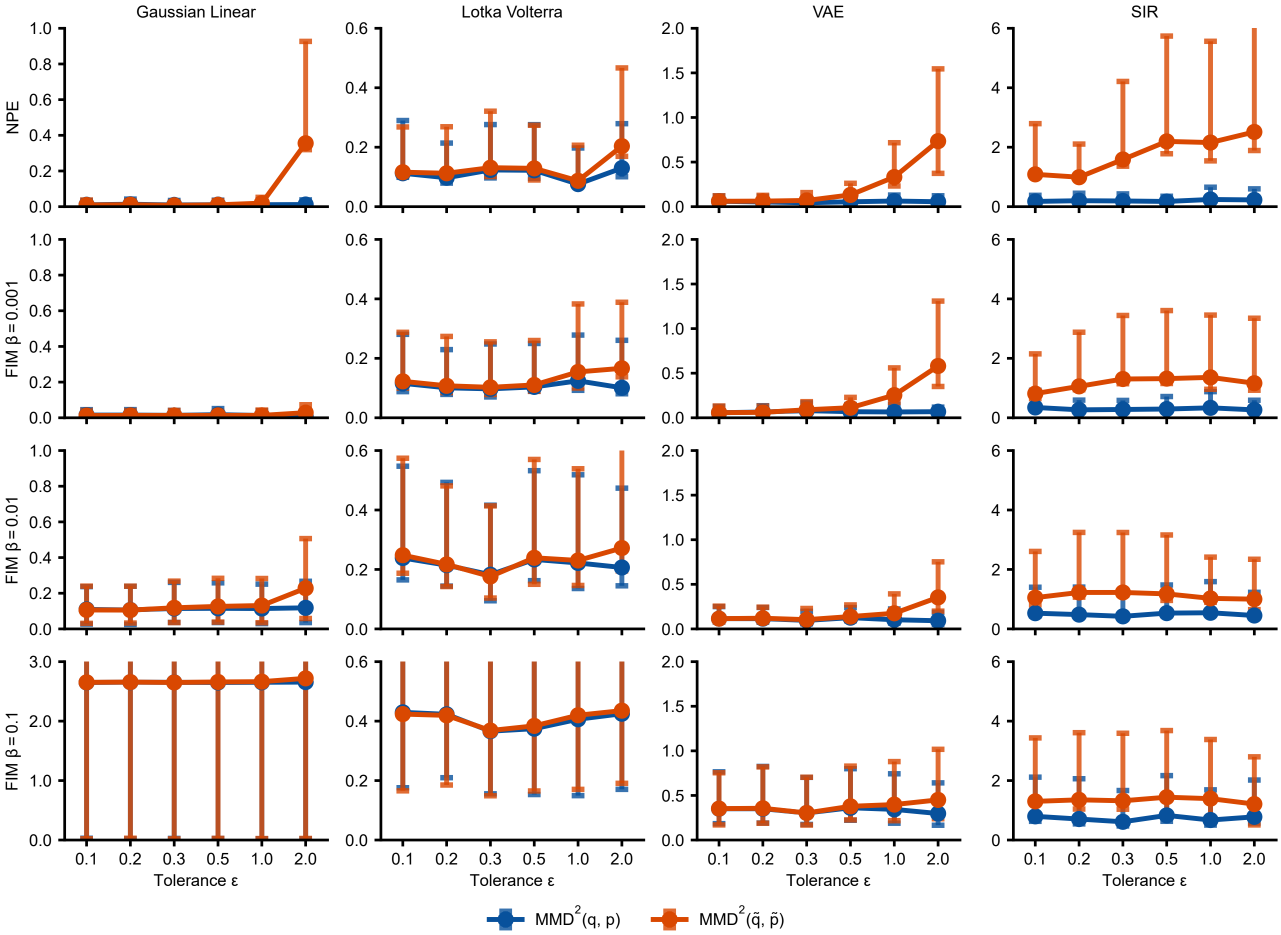

Figure A15. **MMD distance against MCMC-based posteriors on clean and perturbed data.**. This figure showcases the squared MMD distance between the estimated posterior and the posterior obtained through MCMC. The blue line represents the MMD distance on well-specified data, while the orange line illustrates the MMD distance on adversarially perturbed observations. These metrics were computed from a randomly selected subset of 100 clean and perturbed observations.