Robust Classification and Purification with a Single Tensor Network Born Machine

Martin L. Nissen Gonzalez[1⋆] **José Ramón Pareja Monturiol**[2]

[1]Department of Physics and Astronomy, Heidelberg University, 69120 Heidelberg, Germany
[2]Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, 28660 Madrid, Spain
⋆ml-nissen-gonzalez@pm.me

Abstract

Adversarial robustness is closely connected to how well classifiers capture the structure of the data distribution, motivating renewed interest in generative classifiers. We study this connection for matrix product state Born machines, tensor network models whose tractable normalization makes the learned joint density directly accessible. This tractability enables two things at once: detection and purification of adversarial inputs from the model's own density, and a continuous interpolation between discriminative and generative training that injects generative regularization into a classifier with no architectural changes. We propose two purification strategies that reuse the same trained model. Across a synthetic spirals dataset and a larger-scale MNIST benchmark, even a small generative component improves the learned density and strengthens density-based defenses over purely discriminative training. These results are a proof of principle that tensor network generative classifiers can unify robust classification, likelihood estimation, and purification within a single tractable model.

1 INTRODUCTION

Classifiers that achieve high accuracy on natural data can be fooled by imperceptibly small perturbations of their inputs [Goodfellow et al., 2015, Madry et al., 2018]. Adversarial training [Madry et al., 2018] remains the dominant empirical defense, yet it is expensive, reduces clean accuracy, and targets raw robustness without providing density information useful for detection or purification. A complementary direction asks whether building generative structure into the classifier can help: robust classifiers encode useful generative information [Santurkar et al., 2019], and adversarial training has been related to energy-based generative learning [Zhu et al., 2021, Beadini and Masi, 2023]. These observations motivate classifiers that simultaneously learn a joint density over inputs and labels.

Tensor networks (TNs) are a natural fit for this goal. Among them, matrix product states (MPS) [Pérez-García et al., 2007] combine expressive low-rank structure with efficient contraction algorithms and have been used for supervised classification [Stoudenmire and Schwab, 2016] and generative modeling [Han et al., 2018]. Using the Born rule, squared amplitudes define a joint density $p_T(x, c)$ over inputs and labels whose partition function is tractable with isometric embeddings, avoiding the approximate inference required by energy-based generative classifiers. From this one joint we obtain everything we use: conditioning on the input gives the classifier $p_T(c\,|\,x)$, while marginalizing over labels, exactly and efficiently by tensor contraction, gives the input density $p_T(x)$ that drives detection and purification. Existing generative defenses get less from one model: purification methods pair a classifier with a separate generative model [Song et al., 2018, Nie et al., 2022], so the density guiding purification is not the one behind the decision, and energy-based classifiers unify the two but cannot tractably normalize the input density, leaving the interpolation-as-defense regime we study closed to them.

In this work we ask whether MPS Born machines (MPS-BMs) as tractable probabilistic models can improve robustness. Our contributions are scoped to the MPS-BM framework for adversarial robustness, where we combine several individually known ingredients within a single model. First, we adapt the classical discriminative-to-generative interpolation [Bishop and Lasserre, 2007, Yakhnenko et al., 2005] to MPS-BMs, changing only the loss function: $L_\alpha = (1-\alpha)L_D + \alpha L_G$, where L_D and L_G are the conditional and joint negative log-likelihoods. Such a tunable objective is not new for tractable probabilistic models: Peharz et al. [2020] train sum-product networks with the same trade-off, using the resulting density for out-of-distribution detection and calibration. Our question is what that density buys under *adversarial* perturbation: we use $p_T(x)$ to flag low-likelihood inputs, and we propose two purification strategies, likelihood-based and sampling-based, that reuse the same model rather than a second one.

Related work. Interpolating between discriminative and generative objectives dates back to Yakhnenko et al. [2005] and Bishop and Lasserre [2007], and is established for tractable probabilistic models [Peharz et al., 2020]. Score-based classifiers [Zimmermann et al., 2021] and diffusion classifiers [Chen et al., 2024] have pursued generative classifiers for robustness, with mixed success under adaptive attacks [Tramer et al., 2020]. Joint energy-based models target the same accuracy–robustness–generation balance at far larger image scale, but without an exactly normalized input density [Jiang et al., 2026]. Purification defenses so far rely on a second generative model: a PixelCNN in PixelDefend [Song et al., 2018], a diffusion model in DiffPure [Nie et al., 2022]. In the TN literature, MPS generative classifiers have been studied with per-class models [Sun et al., 2020] and GAN-style simultaneous classification–generation [Mossi et al., 2025]. What we add is not the tunable objective but its use for adversarial defense: one model that classifies, detects, and purifies, with exact likelihoods and closed-form conditionals throughout (Appendices A and B).

2 MPS BORN MACHINES AND ROBUST CLASSIFICATION

2.1 MPS BORN MACHINES

Following Stoudenmire and Schwab [2016], we embed each input $x = (x_1, \dots, x_n) \in [-1, 1]^n$ feature-wise via a local map $\phi : [-1, 1] \to \mathbb{C}^d$, forming the global feature vector $\Phi(x) = \bigotimes_{i=1}^{n} \phi(x_i)$. For a K-class problem we include the label through a one-hot vector $e_c \in \mathbb{C}^K$ and define the class-dependent amplitude

$$\psi_T(x, c) = \langle \Phi(x) \otimes e_c,\, T \rangle, \tag{1}$$

where $T \in (\mathbb{C}^d)^{\otimes n} \otimes \mathbb{C}^K$ is a weight tensor parametrized as a matrix product state (MPS): one core per input feature, plus a class core carrying the K-dimensional label leg placed at the centre of the chain (Appendix D). Applying the Born rule gives a joint density over inputs and labels:

$$p_T(x, c) = \frac{|\psi_T(x, c)|^2}{Z_T},$$
$$Z_T = \sum_{c=1}^{K} \int |\psi_T(x, c)|^2 \, dx. \tag{2}$$

The key practical advantage is tractability of Z_T. If ϕ satisfies $\int_{-1}^{1} \overline{\phi_i(x)} \phi_j(x) \, dx = \delta_{ij}$ (isometry), then $Z_T = \|T\|_F^2$, which can be computed efficiently by tensor contraction for MPS. We use the normalized Legendre embedding $\phi_k(x) = \sqrt{(2k+1)/2}\, P_k(x)$, $k = 0, \dots, d-1$, where P_k is the Legendre polynomial of degree k.

The same model serves two purposes. Conditioning on the input gives the classifier $p_T(c \mid x) \propto |\psi_T(x, c)|^2$. Marginalizing over labels gives the input density $p_T(x) = \sum_c p_T(x, c)$, which can be used for detection and purification. Efficient MPS contractions also provide conditional marginals, enabling direct sampling via the product rule.

2.2 DISCRIMINATIVE-TO-GENERATIVE INTERPOLATION

Given samples $(x, c) \sim p_{\text{data}}$, the model can be trained with the discriminative loss $L_D(T) = -\mathbb{E}_{p_{\text{data}}} \log p_T(c \mid x)$ or the generative loss $L_G(T) = -\mathbb{E}_{p_{\text{data}}} \log p_T(x, c)$. Following Bishop and Lasserre [2007], we interpolate with a parameter $\alpha \in [0, 1]$:

$$L_\alpha(T) = (1 - \alpha) L_D(T) + \alpha L_G(T). \tag{3}$$

Substituting the Born rule and the isometric normalization, this becomes

$$\begin{aligned} L_\alpha(T) = & -\mathbb{E}_{p_{\text{data}}} \log |\psi_T(x, c)|^2 \\ & + (1 - \alpha)\, \mathbb{E}_{p_{\text{data}}} \log \sum_{c'} |\psi_T(x, c')|^2 \\ & + \alpha \log \|T\|_F^2. \end{aligned} \tag{4}$$

The first term rewards correct-label amplitude; the second (discriminative) normalizes over classes; the third (generative) normalizes over input space. All three terms are computable in $O((nd + K)r^2)$ per sample with the global norm evaluated once per step. Increasing α adds generative regularization with no architectural change.

2.3 TRAINING

Rather than the DMRG-style sweeps common in tensor-network optimization, we update all cores simultaneously by gradient descent (Adam) on L_α. During training the global norm $Z_T = \|T\|_F^2$ can drift over many orders of magnitude, which is numerically unstable. We therefore add a soft norm regularizer $\lambda\, (\ln c - \ln Z_T)^2$ that pulls the log-norm toward a target $\ln c$. The strength λ is selected by hyperparameter optimization together with the learning rate.

2.4 DETECTION AND PURIFICATION

Detection. Since $p_T(x)$ is available, we flag inputs whose density falls below the q-th percentile of validation likelihoods. This gives a simple, training-free detector for adversarial and out-of-distribution inputs.

Purification. We consider two strategies that use the same trained model. *Likelihood-based purification* solves $\hat{x} \in \arg\min_{x' \in B_\delta^{(\infty)}(\tilde{x}) \cap [-1,1]^n} -\log p_T(x')$ by PGD, moving the perturbed input toward a nearby high-density region. *Sampling-based purification* resamples each coordinate in turn from its model conditional restricted to the ℓ_∞ ball, $\hat{x}_i \sim p_T(x_i \,|\, \hat{x}_1, \dots, \hat{x}_{i-1}, \tilde{x}_{i+1}, \dots, \tilde{x}_n)$ truncated to $x_i \in [\tilde{x}_i - \delta, \tilde{x}_i + \delta]$. This keeps the purified sample within the ℓ_∞ ball of radius δ. The MPS contractions give each conditional

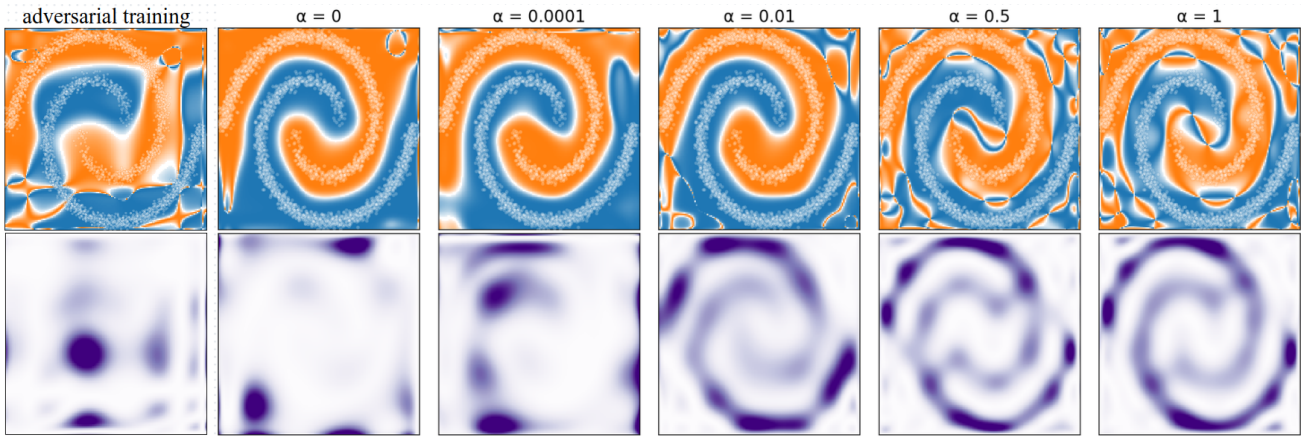


Figure 1: Effect of the interpolation parameter α on the learned classifier and density. *Top row:* conditional distribution $p_T(c|x)$; white marks the decision boundary. *Bottom row:* marginal input density $p_T(x)$. The leftmost column shows an adversarially trained model. Purely discriminative training yields an accurate boundary but poor density; increasing α makes the density concentrate along the spiral geometry.

in closed form, and one or more passes over the coordinates yield the purified input.

3 EXPERIMENTS

3.1 SETUP

We evaluate on two datasets. The first is a two-dimensional spirals dataset with 4,000 points scaled to $[-1, 1]^2$, a controlled benchmark that lets us directly inspect decision boundaries and learned densities (Legendre embedding with $d = 10$, uniform MPS rank $r = 6$, and real-valued cores). The second is MNIST, downscaled to $12 \times 12 = 144$ pixels and embedded feature-wise with a Legendre embedding of dimension $d = 3$ and uniform rank $r = 20$, here with complex-valued cores; we sweep the interpolation parameter over $\alpha \in \{0, 0.01, 0.1, 0.2, 0.5, 1.0\}$. All models use Adam with early stopping on a validation split; the learning rate and remaining hyperparameters, including the norm-regularization strength λ for MNIST, are tuned per architecture and training regime, with model selection on the validation set. Reported metrics are computed on a held-out test set that is not used for model selection. Splits, optimization, and initialization details are in Appendix E.

All perturbations and defenses use the ℓ_∞ norm. Adversarial examples are generated by PGD minimizing the conditional log-likelihood of the true class inside an ℓ_∞ ball of radius ε, evaluated at $\varepsilon \in \{0.1, 0.2, 0.3\}$ (relative strengths $0.05, 0.1, 0.15$ on the rescaled domain). The attacker is white-box on the classifier but unaware of the defenses: the attack is computed once against $p_T(c \mid x)$, and detection or purification is applied afterwards. Purified accuracies are therefore upper bounds rather than robustness claims, which would require a purifier-aware attack [Athalye et al., 2018]; see Appendix C. Purification uses the same ℓ_∞ geometry but with a fixed radius δ equal to 0.1 times the domain size (here $\delta = 0.2$), independent of the attack radius ε. As a baseline we also train adversarially: on spirals with a scheduled radius increasing to $\varepsilon = 0.15$, and on MNIST at relative attack strength 0.3. We report clean accuracy (Clean), accuracy on adversarial examples without defense (Rob.), detection rate and accepted-sample accuracy (Accept.) for likelihood detection at the q-th percentile (defined in Appendix E), and accuracy after likelihood-based (Purif. lk.) and, on spirals, sampling-based (Purif. samp.) purification. The code is available at `https://github.com/MLNissenGonzalez/bm4tc`.

3.2 EFFECT OF GENERATIVE INTERPOLATION

Figure 1 shows how α affects the learned classifier and density on the spirals data, where the density can be visualized directly. The purely discriminative model ($\alpha = 0$) learns a sharp, accurate decision boundary, but its input density hardly resembles p_{data}: conditional training alone gives little incentive to match it. Even so, it still benefits from purification (Section 3.3), indicating that the input distribution is partly learned. Adding even a small generative component quickly concentrates $p_T(x)$ along the spiral geometry. The adversarially trained model is intermediate: its decision regions away from the data manifold resemble the $\alpha > 0$ models more than the discriminative one, yet from $p_T(x)$ alone it is not obvious that it captures p_{data} any better.

To test whether these effects persist in a harder setting, we turn to MNIST, whose 144-dimensional density cannot be visualized directly. The bond dimension we use ($r = 20$) is far below the values up to 800 used to model MNIST generatively with MPS-BMs [Han et al., 2018], so we do not expect the model to capture p_{data} fully. Figure 2 reports accuracy as a function of α; the corresponding negative log-likelihoods are deferred to Appendix F. The signature of

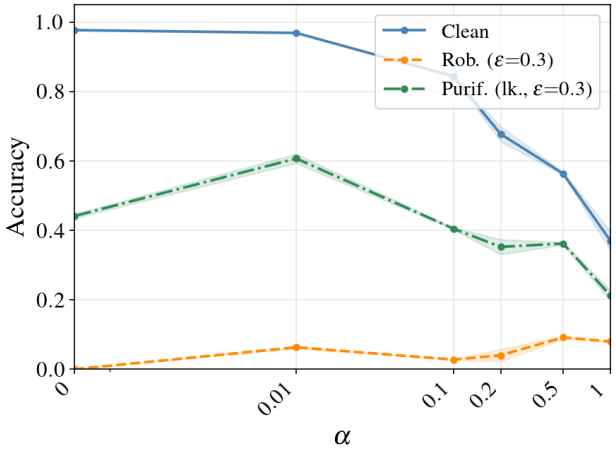


Figure 2: Clean accuracy and, at $\varepsilon = 0.3$, raw robust accuracy and likelihood-purified accuracy on MNIST as functions of α. Classification accuracy drops off beyond $\alpha \approx 0.01$, reflecting underfitting at rank $r = 20$.

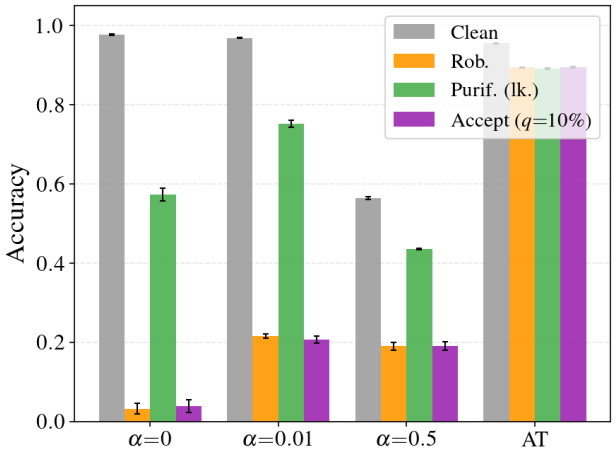


Figure 3: Comparison of training regimes on MNIST at $\varepsilon = 0.2$. Bars show clean, raw robust, likelihood-purified, and accepted-sample accuracies for discriminative ($\alpha = 0$), mixed ($\alpha = 0.01, 0.5$), and adversarially trained (AT) models. Purification gains exceed detection gains, but vanish for adversarial training.

this underfitting is the drop-off in clean accuracy for larger α: beyond $\alpha \approx 0.01$, weighting the generative term more heavily markedly degrades classification. Nonetheless, models with $\alpha > 0$ achieve higher robust accuracy at large ε, and the smallest generative component, $\alpha = 0.01$, outperforms all other interpolated models across clean accuracy, raw robustness, and purification.

3.3 COMPARISON OF TRAINING REGIMES

Figure 3 compares the training regimes on MNIST at $\varepsilon = 0.2$, and two effects stand out. First, purification is far more effective than detection: likelihood-based purification raises robust accuracy from near zero to 0.57 for $\alpha = 0$ and from 0.22 to 0.75 for $\alpha = 0.01$, whereas accepting only inputs that pass the likelihood detector barely improves over no defense. Second, generative regularization helps: the $\alpha = 0.01$ model achieves the best clean–robust balance among the Born machines, while the strongly generative $\alpha = 0.5$ model already pays for underfitting with much lower clean accuracy (0.56). Adversarial training is the strongest baseline by a wide margin: it attains high raw robust accuracy with clean accuracy (0.95) just below the $\alpha = 0.01$ model, and gains essentially nothing from purification, indicating that its robustness comes from reshaped decision boundaries rather than a learned density.

These results show that the qualitative picture from the spirals data carries over to a larger-scale, underfitting regime: a small generative component improves robustness and, especially, purification over purely discriminative training. The gap to adversarial training is consistent with the bond dimension being too small to model p_{data} well, and we expect the density-based defenses to close this gap as the rank grows toward the values at which MPS-BMs are known to model such data faithfully [Han et al., 2018].

4 CONCLUSION

We have used the tractability of MPS Born machines to study several approaches to adversarial robustness within a single model and architecture: by changing only the loss function, the same MPS-BM provides classification, a learned density for detection, and two purification strategies. On the spirals benchmark, moderate generative interpolation improves the learned density substantially while preserving clean accuracy, and this density enables stronger detection and purification than adversarial training. On MNIST, where the bond dimension is too small to model the data distribution, a small generative component still improves robustness and purification over purely discriminative training, though adversarial training remains the stronger baseline in this underfitting regime. The two mechanisms look complementary rather than competing: adversarial training reshapes decision boundaries and gains nothing from purification, whereas generative interpolation builds the density that detection and purification exploit, which is what makes training generatively on adversarial data (below) a natural way to combine them. Two caveats bound them. Our attacks never pass through the purifier, so the purified accuracies are upper bounds until a purifier-aware evaluation [Athalye et al., 2018] is run; and the interpolation is shared with tractable probabilistic models generally [Peharz et al., 2020, Loconte et al., 2025], making density-based defenses a question for that class, not for MPS alone. Scaling the bond dimension toward the values at which generative modeling with MPS is known to be effective [Han et al., 2018] is the natural next step. Another is to train generatively on augmented data, for example noisy or adversarial inputs.

Acknowledgements

M. L. Nissen Gonzalez was supported for nine months by an Erasmus+ traineeship grant. J. R. Pareja Monturiol is supported by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 (CEX2023-001347-S, CEX2019-000904-S, CEX2019-000904-S-20-4, PID2023-146758NB-I00), Comunidad de Madrid (TEC-2024/COM-84-QUITEMAD-CM), Universidad Complutense de Madrid (FEI-EU-22-06), and the Ministry for Digital Transformation and of Civil Service of the Spanish Government through the QUANTUM ENIA project call - Quantum Spain project, and by the European Union through the Recovery, Transformation and Resilience Plan - NextGenerationEU within the framework of the Digital Spain 2026 Agenda.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR, 2018. URL https://proceedings.mlr.press/v80/athalye18a.html.

Senad Beadini and Iacopo Masi. Exploring the connection between robust and generative models, 2023. URL https://arxiv.org/abs/2304.04033.

Christopher M Bishop and Julia Lasserre. Generative or discriminative? getting the best of both worlds. In J M Bernardo, M J Bayarri, J O Berger, A P Dawid, D Heckerman, A F M Smith, and M West, editors, *Bayesian Statistics 8: Proceedings of the Eighth Valencia International Meeting June 2–6, 2006*. Oxford University Press, 07 2007. ISBN 9780199214655. doi: 10.1093/oso/9780199214655.003.0001. URL https://academic.oup.com/book/0/chapter/422208897/chapter-pdf/52446909/isbn-9780199214655-book-part-1.pdf.

Huanran Chen, Yinpeng Dong, Zhengyi Wang, Xiao Yang, Chengqi Duan, Hang Su, and Jun Zhu. Robust classification via a single diffusion model. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pages 6643–6665. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24k.html.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. URL https://arxiv.org/abs/1412.6572.

Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8(3), July 2018. ISSN 2160-3308. doi: 10.1103/physrevx.8.031012. URL http://dx.doi.org/10.1103/PhysRevX.8.031012.

Kaichao Jiang, He Wang, Xiaoshuai Hao, Xiulong Yang, Ajian Liu, Qi Chu, Yunfeng Diao, and Richang Hong. Your classifier can do more: Towards balancing the gaps in classification, robustness, and generation, 2026. URL https://arxiv.org/abs/2505.19459.

Lorenzo Loconte, Antonio Mari, Gennaro Gala, Robert Peharz, Cassio de Campos, Erik Quaeghebeur, Gennaro Vessio, and Antonio Vergari. What is the relationship between tensor factorizations and circuits (and how can we exploit it)? *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2409.07953.

Radek Mackowiak, Lynton Ardizzone, Ullrich Köthe, and Carsten Rother. Generative classifiers as a basis for trustworthy image classification. In *CVPR*, pages 2970–2980, 2021. doi: 10.1109/CVPR46437.2021.00299. URL https://arxiv.org/abs/2007.15036.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Alex Meiburg, Jing Chen, Jacob Miller, Raphaëlle Tihon, Guillaume Rabusseau, and Alejandro Perdomo-Ortiz. Generative learning of continuous data by tensor networks. *SciPost Phys.*, 18(3):096, 2025. doi: 10.21468/SciPostPhys.18.3.096.

Joshua B. Moore, Hugo P. Stackhouse, Ben D. Fulcher, and Sahand Mahmoodian. Using matrix-product states for time-series machine learning. *Phys. Rev. Res.*, 7 (4), October 2025. ISSN 2643-1564. doi: 10.1103/61h8-8qr5. URL http://dx.doi.org/10.1103/61h8-8qr5.

Alex Mossi, Bojan Žunkovič, and Kyriakos Flouris. A matrix product state model for simultaneous classification and generation. *Quantum Mach. Intell.*, 7 (1):48, 2025. ISSN 2524-4914. doi: 10.1007/s42484-025-00272-6. URL https://doi.org/10.1007/s42484-025-00272-6.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification, 2022. URL https://arxiv.org/abs/2205.07460.

Alexander Novikov, Mikhail Trofimov, and Ivan V. Oseledets. Exponential machines. *Bull. Pol. Acad.*

Sci. Tech. Sci., 66(No 6 (Special Section on Deep Learning: Theory and Practice)):789–797, 2018. doi: 10.24425/bpas.2018.125926. URL https://journals.pan.pl/dlibra/publication/125926/edition/109868/content.

José Ramón Pareja Monturiol, David Pérez-García, and Alejandro Pozas-Kerstjens. TensorKrowch: Smooth integration of tensor networks in machine learning. *Quantum*, 8:1364, 2024. doi: 10.22331/q-2024-06-11-1364. URL https://arxiv.org/abs/2306.08595. https://github.com/joserapa98/tensorkrowch.

Robert Peharz, Antonio Vergari, Karl Stelzner, Alejandro Molina, Xiaoting Shao, Martin Trapp, Kristian Kersting, and Zoubin Ghahramani. Random sum-product networks: A simple and effective approach to probabilistic deep learning. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 334–344. PMLR, 2020. URL https://proceedings.mlr.press/v115/peharz20a.html.

David Pérez-García, Frank Verstraete, Michael M. Wolf, and J. Ignacio Cirac. Matrix product state representations. *Quantum Inf. Comput.*, 7(5):401—430, July 2007. ISSN 1533-7146. doi: 10.26421/QIC7.5-6-1. URL https://arxiv.org/abs/0608197.

Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/6f2268bd1d3d3ebaabb04d6b5d099425-Paper.pdf.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJUYGxbCW.

Edwin Stoudenmire and David J. Schwab. Supervised learning with tensor networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *NeurIPS*, volume 29, pages 4799–4807. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5314b9674c86e3f9d1ba25ef9bb32895-Abstract.html.

Zheng-Zhi Sun, Cheng Peng, Ding Liu, Shi-Ju Ran, and Gang Su. Generative tensor network classification model for supervised machine learning. *Phys. Rev. B*, 101:075135, Feb 2020. doi: 10.1103/PhysRevB.101.075135. URL https://link.aps.org/doi/10.1103/PhysRevB.101.075135.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *NeurIPS*, volume 33, pages 1633–1645. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/11f38f8ecd71867b42433548d1078e38-Paper.pdf.

Jinhui Wang, Chase Roberts, Guifré Vidal, and Stefan Leichenauer. Anomaly detection with tensor networks, 2020. URL https://arxiv.org/abs/2006.02516.

Oksana Yakhnenko, Adrian Silvescu, and Vasant Honavar. Discriminatively trained markov model for sequence classification. In *Fifth IEEE International Conference on Data Mining (ICDM)*, pages 498–505. IEEE, 2005.

Yao Zhu, Jiacheng Ma, Jiacheng Sun, Zewei Chen, Rongxin Jiang, Yaowu Chen, and Zhenguo Li. Towards understanding the generative capability of adversarially robust classifiers. In *ICCV*, pages 7708–7717, 2021. doi: 10.1109/ICCV48922.2021.00763. URL https://arxiv.org/abs/2108.09093.

Roland S. Zimmermann, Lukas Schott, Yang Song, Benjamin A. Dunn, and David A. Klindt. Score-based generative classifiers, 2021. URL https://arxiv.org/abs/2110.00473.

Robust Classification and Purification with a Single Tensor Network Born Machine (Supplementary Material)

Martin L. Nissen Gonzalez[1⋆] **José Ramón Pareja Monturiol**[2]

[1]Department of Physics and Astronomy, Heidelberg University, 69120 Heidelberg, Germany
[2]Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, 28660 Madrid, Spain
⋆ml-nissen-gonzalez@pm.me

A RELATED WORK

Adversarial attacks and defenses. Adversarial examples are small, norm-bounded perturbations that cause misclassification [Goodfellow et al., 2015]. Projected gradient descent (PGD) attacks provide a standard first-order benchmark, and adversarial training against such attacks remains the dominant empirical defense [Madry et al., 2018]. However, robustness must be evaluated against the full defensive pipeline: when a defense includes preprocessing, rejection, or purification, adaptive attacks can be designed to target these components [Tramer et al., 2020]. A complementary class of defenses uses purification, mapping inputs back toward the data manifold before classification using a generative model. PixelDefend [Song et al., 2018] does this with a PixelCNN density model, and diffusion-based purification [Nie et al., 2022] with a diffusion model. Both attach an auxiliary generative model to a separately trained classifier, so the purifier's density and the classifier's decision function are learned independently and need not agree; the setting we study collapses the two into one model (Appendix B).

Generative classifiers and robustness. Interpolating between discriminative and generative objectives to combine their advantages was first used by Yakhnenko et al. [2005] and interpreted within Bayesian machine learning by Bishop and Lasserre [2007]. This perspective has gained renewed relevance in adversarial robustness, where robust classifiers encode useful generative information [Santurkar et al., 2019] and adversarial training has been related to energy-based generative learning [Zhu et al., 2021, Beadini and Masi, 2023]. Modern generative classifiers include score-based classifiers [Zimmermann et al., 2021], normalizing flows [Mackowiak et al., 2021], and diffusion classifiers [Chen et al., 2024]. These works suggest generative structure can help with robustness, OOD detection, and purification, but the picture is not uniformly positive: score-based classifiers remain vulnerable to attacks [Zimmermann et al., 2021]. Closest in spirit to our aim, Jiang et al. [2026] target the same three-way accuracy–robustness–generation balance with a joint energy-based model trained on adversarial as well as clean data, and at far larger image scale than our MPS-BMs reach. The approaches differ in which quantities are available exactly, not in ambition. In a joint energy-based classifier the class conditional $p_T(c\,|\,x)$ is normalized over the K classes and is tractable, but the partition function over input space is not, so an exactly normalized input density and closed-form input conditionals are out of reach. In the MPS-BM both are available: $p_T(x)$ is exactly normalized by the contractible $Z_T = \|T\|_F^2$, and every conditional $p_T(x_i\,|\,x_{\neq i})$ is closed-form. Our sampling-based purification and the joint-likelihood attack of Appendix J rely on exactly these quantities (see also Appendix B).

Tractable probabilistic models. MPS-BMs are not the only model class with an exactly normalized density, and the discriminative-to-generative interpolation we use is established elsewhere in that class. Most directly, Peharz et al. [2020] train random sum-product networks (RAT-SPNs) with the hybrid objective $H = \lambda\,\mathrm{CE} - (1-\lambda)\,\mathrm{LL}/|X|$, recovering purely discriminative training at $\lambda = 1$ and purely generative training at $\lambda = 0$; this is our L_α under the identification $\alpha = 1 - \lambda$. Their evaluation targets out-of-distribution detection, calibration, and robustness to missing features rather than adversarial perturbations, and they do not consider purification. We therefore do not claim the interpolation as novel: our question is what the resulting density is worth against an adversary, and whether one tractable model can serve as classifier, detector, and purifier at once. This framing is not specific to MPS. Squared tensor-network factorizations and squared probabilistic circuits are the same objects viewed from two literatures [Loconte et al., 2025], so an MPS-BM is a particular squared circuit, and the defenses we study transfer in principle to probabilistic circuits with an exact partition function. What would

change is the inductive bias rather than the mechanism, which makes a circuit-side replication the natural test of how much of our result is about the density and how much about the MPS chain.

Tensor networks for machine learning. MPS have been used for supervised classification [Stoudenmire and Schwab, 2016, Novikov et al., 2018] and generative modeling [Han et al., 2018]. MPS generative classifiers have been studied by training per-class conditioned models [Sun et al., 2020] or with a GAN-style simultaneous objective [Mossi et al., 2025]; robustness to random noise has been reported for the latter. Within the MPS-BM framework, Moore et al. [2025] leverage the inductive bias of tensor networks for time-series classification and imputation under a single logarithmic loss, and Meiburg et al. [2025] extend Born-machine generative modeling to continuous data. TNs have also shown promise in anomaly detection [Wang et al., 2020]. Our work complements these by exploiting the density of a single interpolated MPS-BM for adversarial detection and purification, a use none of them consider.

B WHY BORN MACHINES?

A reasonable objection is that deep generative models already perform detection and purification, PixelDefend [Song et al., 2018] being the canonical example, and at far higher fidelity on image data than an MPS-BM of the ranks we use. We agree, and we do not present these results as competitive image defenses. The properties that motivate the MPS-BM here are structural rather than quantitative. Above all, it is a minimal instrument for the question we ask: the classifier, the detector's likelihood, and the purifier's objective are one set of parameters, and the only moving part is the loss weight α. This lets us study the interaction between adversarial robustness and generative modeling with far fewer confounds than a pipeline of separately trained models, since what varies across our comparisons is α alone rather than which of several networks was tuned.

One model rather than two. PixelDefend purifies with a PixelCNN and classifies with a separate network; DiffPure [Nie et al., 2022] pairs a diffusion model with a separate classifier. The density guiding purification is then not the density underlying the decision, and the defense inherits a second model to train, tune, and defend. In an MPS-BM the classifier, the detector's likelihood, and the purifier's objective are three views of the same parameters, and α tunes all three at once.

Exactness. The partition function is computable by contraction, so $\log p_T(x)$ is exact rather than bounded or estimated, and every conditional $p_T(x_i \mid x_{\neq i})$ is available in closed form. This is what makes sampling-based purification exact restricted-conditional resampling rather than an approximate MCMC scheme, and it lets us attribute changes in the defenses to α instead of to inference error.

Where this does and does not apply. The honest limitation is the inductive bias: an MPS imposes a one-dimensional chain over a flattened pixel grid, which is a poor match for images, and the rank-limited underfitting we report on MNIST (Section 3.2) is a direct consequence. MNIST here is a controlled testbed, not a domain claim; its value is that exact likelihoods isolate the effect of the interpolation. The chain structure is natural for sequential data, where MPS-BMs are effective at classification and imputation [Moore et al., 2025, Meiburg et al., 2025], and we expect that to be the setting where these defenses are worth their cost. Identifying time-series threat models in which adversarial perturbation is a realistic concern, rather than transplanting an image-derived ℓ_∞ threat model, is the open question there.

C THREAT MODEL AND LIMITATIONS

Threat model. The attacker has white-box access to the trained model and computes an ℓ_∞-bounded perturbation of radius ε by PGD on the conditional $p_T(c\,|\,x)$. The attacker is *not* aware of the defenses: adversarial examples are generated once against the classifier, and detection or purification is applied to those fixed inputs afterwards. The purification radius δ and the detection threshold q are chosen by the defender and not treated as known to the attacker.

The purifier is outside the attacked pipeline. This is the principal limitation of our evaluation, and it applies to every purified number we report. The standard we describe in Appendix A, that a defense be evaluated against an attacker who targets the full pipeline [Tramer et al., 2020], is one our purification results do not meet: an attacker who knows purification will follow can search for perturbations that survive it, and non-adaptive evaluations of preprocessing defenses have repeatedly proved optimistic [Athalye et al., 2018]. Our purified accuracies should therefore be read as upper bounds under a purifier-unaware attacker, not as robust accuracy. Closing this gap requires backpropagating through the defense. Likelihood-based purification is itself a PGD loop and could in principle be unrolled at considerable cost, while sampling-based purification is non-differentiable, so BPDA [Athalye et al., 2018], which approximates the purifier's backward pass by the identity, is the tractable route for both. We have not run this evaluation, and we flag it as the necessary next step rather

than a detail. The joint-likelihood attack of Appendix J is a partial step: it is aware of the *density* that the detector uses, but it still does not attack through the purifier.

Other limitations. The bond dimension on MNIST is far below what is needed to model p_{data} faithfully [Han et al., 2018], so our MNIST results characterize an underfitting regime and the comparison against adversarial training should be read in that light. We also report no probabilistic-circuit baseline, which the correspondence of Loconte et al. [2025] would make a natural control for separating the contribution of the exact density from that of the MPS inductive bias.

D MPS DECOMPOSITION

The weight tensor $T \in (\mathbb{C}^d)^{\otimes n} \otimes \mathbb{C}^K$ of Eq. (1) has $d^n K$ entries and is never formed explicitly. Instead it is parametrized as a matrix product state (MPS), a tensor train of low-rank cores: n feature cores and a single class core. Each interior feature core $A^{(i)} \in \mathbb{C}^r \otimes \mathbb{C}^d \otimes \mathbb{C}^r$ carries one physical leg of dimension d, contracted with the local embedding $\phi(x_i)$, and two bond legs of dimension r. The two boundary cores are rank-two tensors, $A^{(1)} \in \mathbb{C}^d \otimes \mathbb{C}^r$ and $A^{(n)} \in \mathbb{C}^r \otimes \mathbb{C}^d$, each carrying a single bond leg. The class core $A^{(\mathrm{c})} \in \mathbb{C}^r \otimes \mathbb{C}^K \otimes \mathbb{C}^r$ carries the physical leg of dimension K, contracted with the one-hot label e_c, and is placed at the centre of the chain. All bond dimensions are set to a single uniform rank r. The decomposition expresses each entry of T as a sum over the bond indices $\{a\}$,

$$T^{c}_{i_1 \cdots i_n} = \sum_{\{a\}} A^{(1)}_{i_1 a_1} A^{(2)}_{a_1 i_2 a_2} \cdots A^{(\mathrm{c})}_{a_{m-1}\, c\, a_m} \cdots A^{(n)}_{a_{n-1} i_n}, \tag{5}$$

with the class core inserted at the centre of the feature chain.

The amplitude $\psi_T(x, c)$ is obtained by contracting this decomposition with the input and label: each feature core is contracted on its physical leg with the embedding vector $\phi(x_i)$ and the class core with e_c, turning every core into a matrix, and the resulting chain of matrices is multiplied along the chain. This never forms T and costs $O(ndr^2)$ per sample; the order in which the bonds are summed is one implementation choice among several.

The discriminative and generative computations differ only in the normalization. Leaving the class leg open, that is, not contracting $A^{(\mathrm{c})}$ with any e_c, produces all K amplitudes $\psi_T(x, c')$ in a single forward pass, from which the conditional classifier $p_T(c \,|\, x)$ follows by squaring and normalizing over the K values. The generative normalization $Z_T = \|T\|_F^2 = \langle T, T \rangle$ is a self-contraction of the MPS with its conjugate, computed once per optimization step. Since one-hot class vectors are orthonormal, the class core also satisfies the isometricity condition, so including it leaves the partition function equal to $\|T\|_F^2$.

E IMPLEMENTATION DETAILS

All experiments were run on an Intel Xeon CPU E5-2620 v4 with 256 GB RAM and an NVIDIA GeForce RTX 3090. We used PyTorch, SciPy, and TensorKrowch [Pareja Monturiol et al., 2024] to implement MPS-BM models. Each dataset is split into train/validation/test partitions with ratios $[0.5, 0.25, 0.25]$ for spirals and $[0.8, 0.1, 0.1]$ for MNIST; the validation split is used for early stopping and model selection and the test split only for the reported metrics. All cores are initialized with the `randn_eye` scheme, which exploits that embeddings such as Legendre place a 1 in their first component; anticipating this, the scheme keeps the model output of order one at initialization. For MNIST we additionally apply the norm regularizer of Section 2.3, with target $\ln c \in \{0,\, n \ln d\}$ or the norm of a pretrained MPS-BM, and tune λ together with the (low) learning rate; the two-dimensional spirals models train stably without it. The MNIST models are also warm-started: we first train the discriminative model ($\alpha = 0$) and initialize the adversarial and $\alpha > 0$ runs from it (this pretrained model also supplies the norm target above), whereas on spirals the $\alpha > 0$ models are trained from scratch. On spirals the interpolated models were evaluated over 20 random seeds at $\alpha \in \{0, 1\}$ and 5 seeds at intermediate values, and adversarial training over 8 seeds due to higher computational cost; on MNIST all models were evaluated over 5 seeds. The PGD-based likelihood purification used 50 steps with step size 10^{-2} and projection onto $B_\delta^{(\infty)}(\tilde{x}) \cap [-1, 1]^n$. For efficiency, sampling-based (Gibbs) purification was applied only to the spirals models and not to MNIST. For likelihood detection we set the threshold to the 10th percentile of validation-set likelihoods, that is, to a 10% false-positive rate on natural data ($q = 10\%$). The detection rate and accepted-sample accuracy are computed on a test set of purely adversarial examples, with no natural samples mixed in: the detection rate is the fraction flagged as below threshold, and the accepted-sample accuracy (Accept.) is the classification accuracy on the adversarial inputs that lie above the threshold and are therefore not rejected. The code is publicly available at `https://github.com/MLNissenGonzalez/bm4tc`.

F ADDITIONAL MNIST RESULTS

Figure 4 reports the discriminative and generative negative log-likelihoods on MNIST as functions of α, the loss-side counterpart to the accuracy curve in the main text.

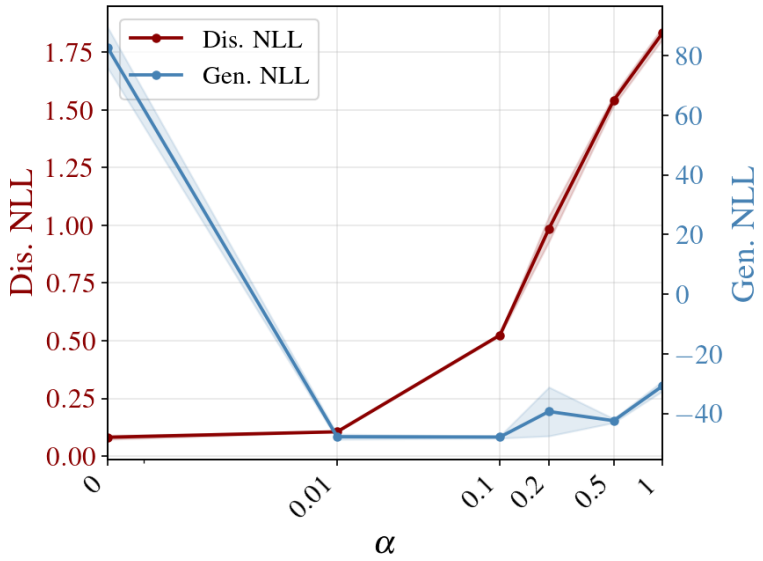
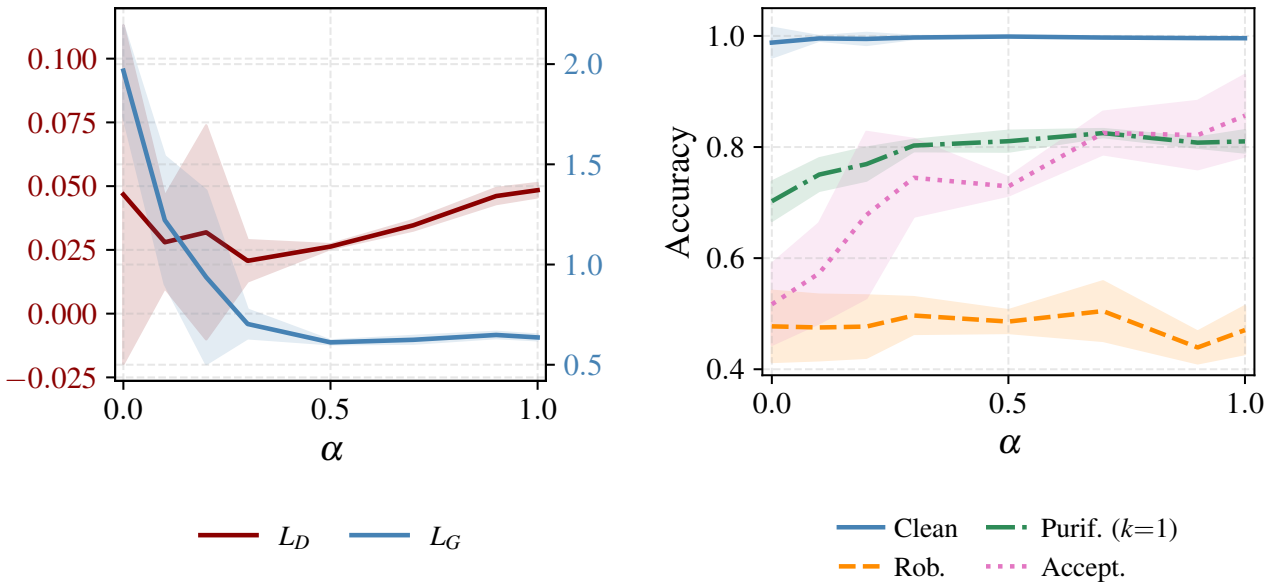


Figure 4: Discriminative (L_D) and generative (L_G) negative log-likelihoods on MNIST as functions of α. Curves are averages over five seeds; shaded regions show $\pm$ one standard deviation. Increasing α lowers the generative loss while raising the discriminative loss, the trade-off underlying the accuracy drop-off in the main text.

G SPIRALS DATASET: QUANTITATIVE RESULTS

On the low-dimensional spirals data the MPS-BM is expressive enough to fit the distribution well, complementing the underfitting MNIST results in the main text. This appendix collects the quantitative spirals results.

Figures 5a and 5b report the negative log-likelihoods and accuracy metrics as functions of α. The discriminative loss is minimized at intermediate α, suggesting that moderate generative interpolation regularizes the model without sacrificing classification. The spirals distribution is simple enough that the model fits it very well, so adding generative pressure costs little clean accuracy, unlike the MNIST underfitting regime, where the generative term trades off against classification. Clean and raw robust accuracy vary only mildly with α; the purely generative model ($\alpha = 1$) shows slightly lower raw robustness, likely because concentrating the density near the manifold makes conditional probabilities in low-likelihood regions less reliable. The main gains appear in the density-based defenses: as α increases, likelihood-based detection becomes more effective and sampling purification achieves higher final accuracy.



(a) Negative log-likelihoods

(b) Accuracy metrics

Figure 5: Effect of generative interpolation on the spirals dataset ($\varepsilon = 0.2$) as functions of α. *(a)* Discriminative (L_D) and generative (L_G) negative log-likelihoods. *(b)* Clean accuracy, raw robust accuracy, sampling-purified accuracy, and accepted-sample accuracy after likelihood detection. Curves are averages over random seeds; shaded regions show $\pm$ one standard deviation.

Figure 6 compares the four training regimes. Clean accuracy is near-perfect in all cases. Adversarial training gives the best raw robust accuracy, but the gain is modest and does not extend to the strongest detection or purification performance. Mixed and generative models, whose likelihoods are trained explicitly, dominate once density-based defenses are applied: the generative model performs best on accepted samples while the mixed model provides the strongest overall balance across all metrics. Sampling-based purification slightly outperforms likelihood-based purification, with a single resampling pass capturing most of the gain.

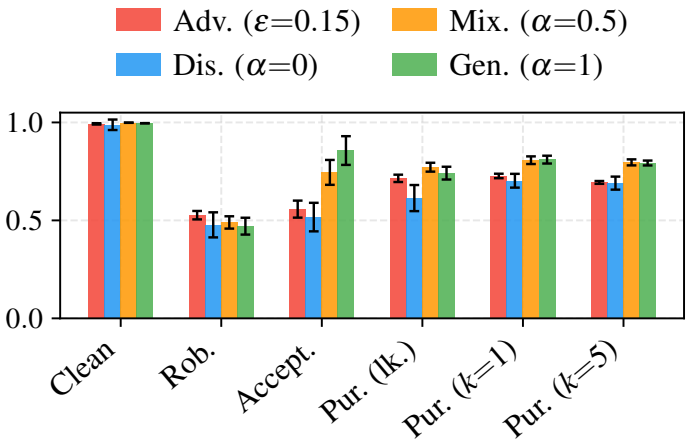


Figure 6: Comparison of training regimes on spirals ($\varepsilon = 0.2$). Bars show clean, robust, accepted, and purified accuracies for adversarial training, discriminative ($\alpha = 0$), mixed ($\alpha = 0.5$), and generative ($\alpha = 1$) models. Purification is evaluated with likelihood-based and sampling-based ($k = 1, 5$ resampling passes) methods.

H ADDITIONAL RESULTS ACROSS ATTACK RADII

Table 1: Summary of results on the spirals dataset for conditional log-likelihood attacks with $\varepsilon \in \{0.1, 0.2, 0.3\}$. We compare adversarial training ($\varepsilon = 0.15$), purely discriminative training ($\alpha = 0$), mixed training ($\alpha = 0.5$), and purely generative training ($\alpha = 1$). The best-performing model for each metric is highlighted in bold. Values are averages over random seeds $\pm$ one standard deviation, using 20 seeds for the interpolated models and 8 seeds for adversarial training. Overall, the mixed model achieves the best performance across the largest number of metrics, while adversarial training performs best for raw robust accuracy under stronger attacks and the purely generative model performs best for adversarial detection rate.

Model	Metric	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.3$
Adv. ($\varepsilon = 0.15$)	Clean		0.993 ± 0.003	
	Rob.	0.748 ± 0.020	$\mathbf{0.527 \pm 0.022}$	$\mathbf{0.417 \pm 0.022}$
	Purif. (lk.)	0.870 ± 0.024	0.715 ± 0.019	0.596 ± 0.037
	Purif. (samp., k=1)	0.847 ± 0.014	0.727 ± 0.011	0.628 ± 0.024
	Det. rate (q=10%)	0.288 ± 0.042	0.323 ± 0.040	0.290 ± 0.026
	Accept. (q=10%)	0.848 ± 0.013	0.557 ± 0.043	0.415 ± 0.041
Disc. ($\alpha = 0$)	Clean		0.988 ± 0.027	
	Rob.	0.751 ± 0.092	0.477 ± 0.064	0.336 ± 0.060
	Purif. (lk.)	0.786 ± 0.068	0.614 ± 0.067	0.511 ± 0.071
	Purif. (samp., k=1)	0.810 ± 0.045	0.702 ± 0.035	0.606 ± 0.034
	Det. rate (q=10%)	0.177 ± 0.036	0.178 ± 0.050	0.164 ± 0.055
	Accept. (q=10%)	0.798 ± 0.095	0.517 ± 0.073	0.353 ± 0.070
Mixed ($\alpha = 0.5$)	Clean		$\mathbf{0.999 \pm 0.001}$	
	Rob.	$\mathbf{0.870 \pm 0.021}$	0.490 ± 0.031	0.373 ± 0.034
	Purif. (lk.)	$\mathbf{0.978 \pm 0.011}$	$\mathbf{0.771 \pm 0.023}$	$\mathbf{0.647 \pm 0.036}$
	Purif. (samp., k=1)	$\mathbf{0.937 \pm 0.012}$	0.807 ± 0.020	$\mathbf{0.712 \pm 0.019}$
	Det. rate (q=10%)	$\mathbf{0.600 \pm 0.038}$	$\mathbf{0.767 \pm 0.034}$	$\mathbf{0.733 \pm 0.028}$
	Accept. (q=10%)	0.981 ± 0.008	0.745 ± 0.063	0.504 ± 0.057
Gen. ($\alpha = 1$)	Clean		0.996 ± 0.001	
	Rob.	0.840 ± 0.035	0.471 ± 0.043	0.356 ± 0.041
	Purif. (lk.)	0.962 ± 0.007	0.741 ± 0.033	0.604 ± 0.034
	Purif. (samp., k=1)	0.932 ± 0.013	$\mathbf{0.810 \pm 0.020}$	0.706 ± 0.022
	Det. rate (q=10%)	0.530 ± 0.055	0.735 ± 0.037	0.719 ± 0.024
	Accept. (q=10%)	$\mathbf{0.993 \pm 0.007}$	$\mathbf{0.857 \pm 0.073}$	$\mathbf{0.593 \pm 0.105}$

Table 2: Summary of MNIST results for conditional log-likelihood attacks with $\varepsilon \in \{0.1, 0.2, 0.3\}$. We compare purely discriminative training ($\alpha = 0$), mixed training ($\alpha = 0.01, 0.5$), and adversarial training (AT, relative strength 0.3). Clean accuracy is attack-independent and reported once per model. The best value for each metric and radius is highlighted in bold. Values are averages over five seeds $\pm$ one standard deviation. Likelihood-based purification is far more effective than detection across all radii, while adversarial training dominates raw robust accuracy.

Model	Metric	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.3$
	Clean		$\mathbf{0.977 \pm 0.001}$	
	Rob.	0.455 ± 0.030	0.032 ± 0.013	0.000 ± 0.000
α=0	Purif. (lk.)	0.815 ± 0.020	0.574 ± 0.016	0.441 ± 0.006
	Det. rate (q=10%)	$\mathbf{0.294 \pm 0.028}$	$\mathbf{0.287 \pm 0.036}$	0.207 ± 0.031
	Accept. (q=10%)	0.554 ± 0.033	0.038 ± 0.016	0.000 ± 0.000
	Clean		0.969 ± 0.000	
	Rob.	0.677 ± 0.007	0.215 ± 0.006	0.063 ± 0.003
α=0.01	Purif. (lk.)	0.883 ± 0.002	0.751 ± 0.009	0.607 ± 0.013
	Det. rate (q=10%)	0.164 ± 0.006	0.275 ± 0.005	$\mathbf{0.367 \pm 0.007}$
	Accept. (q=10%)	0.678 ± 0.005	0.207 ± 0.008	0.080 ± 0.006
	Clean		0.564 ± 0.004	
	Rob.	0.384 ± 0.005	0.190 ± 0.010	0.092 ± 0.004
α=0.5	Purif. (lk.)	0.487 ± 0.003	0.436 ± 0.001	0.362 ± 0.004
	Det. rate (q=10%)	0.096 ± 0.003	0.194 ± 0.004	0.305 ± 0.009
	Accept. (q=10%)	0.385 ± 0.007	0.190 ± 0.010	0.096 ± 0.005
	Clean		0.954 ± 0.001	
	Rob.	$\mathbf{0.930 \pm 0.001}$	$\mathbf{0.894 \pm 0.001}$	$\mathbf{0.848 \pm 0.002}$
AT	Purif. (lk.)	$\mathbf{0.922 \pm 0.002}$	$\mathbf{0.891 \pm 0.002}$	$\mathbf{0.849 \pm 0.002}$
	Det. rate (q=10%)	0.122 ± 0.012	0.158 ± 0.027	0.196 ± 0.034
	Accept. (q=10%)	$\mathbf{0.931 \pm 0.002}$	$\mathbf{0.894 \pm 0.002}$	$\mathbf{0.848 \pm 0.005}$

I EFFECT OF MODEL CAPACITY AND FEATURE EMBEDDING

In this appendix, we analyze how the expressiveness of the MPS-BM affects the learned classifier, density, and robustness metrics. We consider two sources of expressiveness: the MPS capacity, controlled by the local embedding dimension d and rank r, and the choice of feature embedding, which determines the local basis used to represent functions of each input coordinate.

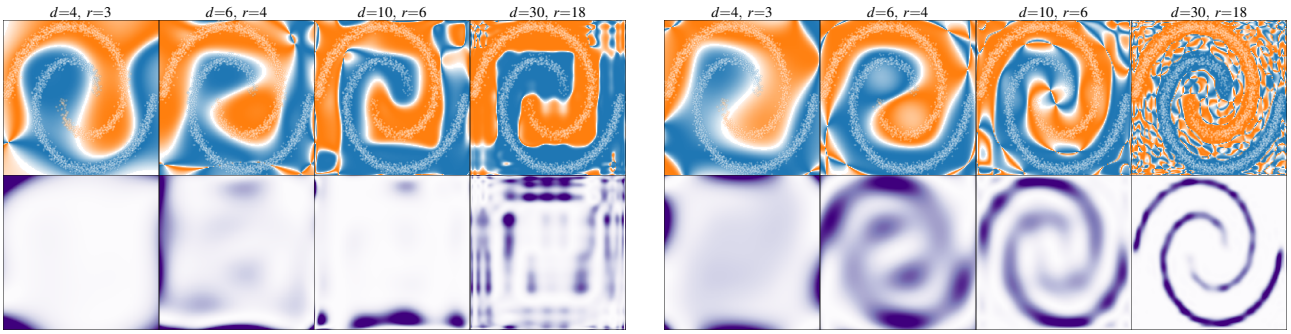
I.1 EFFECT OF MODEL CAPACITY

We first vary the embedding dimension d and rank r for Legendre embeddings. Figure 7 shows the learned decision regions and marginal input densities for discriminative and generative training. Increasing (d, r) makes the model substantially more expressive: the decision boundary becomes better adapted to the spiral geometry, and the learned density becomes increasingly concentrated around the data manifold. This effect is especially clear for the generative model, where the largest setting nearly recovers the spiral density.

The corresponding robustness metrics at $\varepsilon = 0.2$ are shown in Figure 8. Higher-capacity models generally improve the reported accuracies, with the largest gains appearing after purification. For the generative model, this reflects the improved density estimate: once the data manifold is captured more accurately, likelihood-based purification and detection become more effective.

Figure 9 further compares discriminative and generative models across attack radii. As expected, all accuracies decrease as ε increases. The generative models can have lower raw robust accuracy, especially at higher capacity, which is consistent with the behavior observed in the main text: when the density concentrates around the data manifold, classification in low-density regions between classes becomes less stable. However, density-based purification substantially improves performance, particularly for the more expressive generative models. This reinforces that the benefit of the generative component is not primarily raw robustness, but the availability of a useful density for purification.

This capacity dependence puts the MNIST results of the main text in context. With 144 features and rank $r = 20$, the MNIST



(a) Discriminative training ($\alpha = 0$)

(b) Generative training ($\alpha = 1$)

Figure 7: Effect of model capacity on the learned classifier and density for Legendre embeddings. Each panel increases the embedding dimension and rank from left to right, with $(d, r) = (4, 3), (6, 4), (10, 6), (30, 18)$. Top rows show the conditional distribution $p_T(c \mid x)$ as decision regions, with dataset points overlaid in light gray; bottom rows show the marginal input density $p_T(x)$. Increasing capacity improves both the decision boundary and the density estimate, with the largest model capturing the spiral structure most accurately.

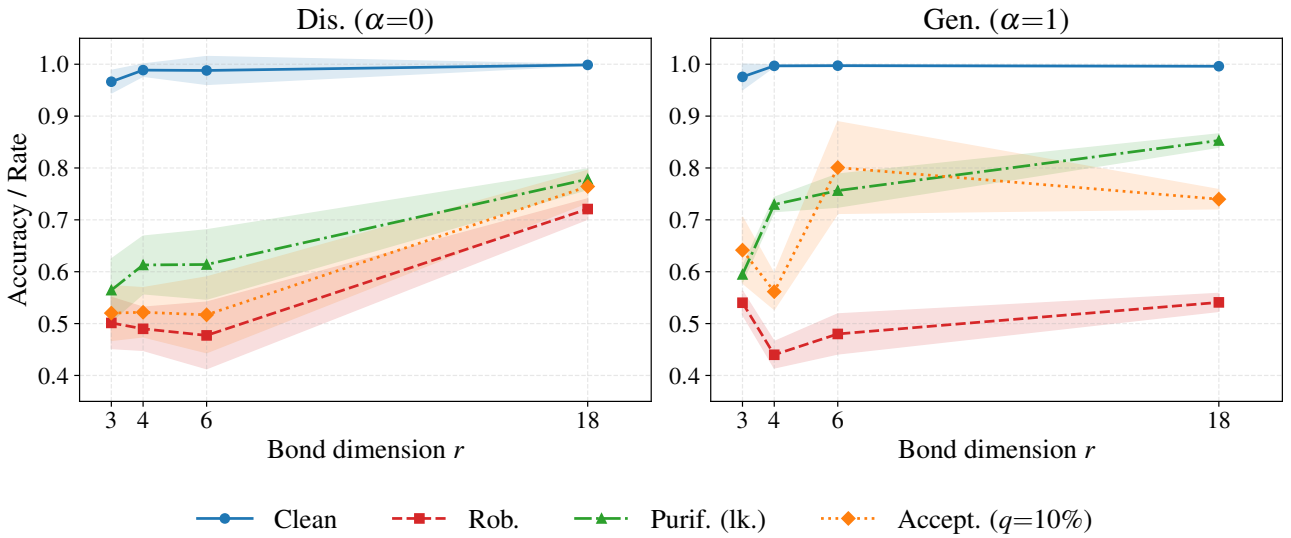


Figure 8: Effect of model capacity on robustness metrics for Legendre embeddings at $\varepsilon = 0.2$. We compare discriminative training ($\alpha = 0$) and generative training ($\alpha = 1$) across increasing pairs $(d, r) = (4, 3), (6, 4), (10, 6), (30, 18)$. Curves show clean accuracy, raw robust accuracy, likelihood-purified accuracy, and accuracy on accepted samples after likelihood-based detection with $q = 10\%$. Larger models achieve better robustness metrics, with the strongest improvement appearing after purification.

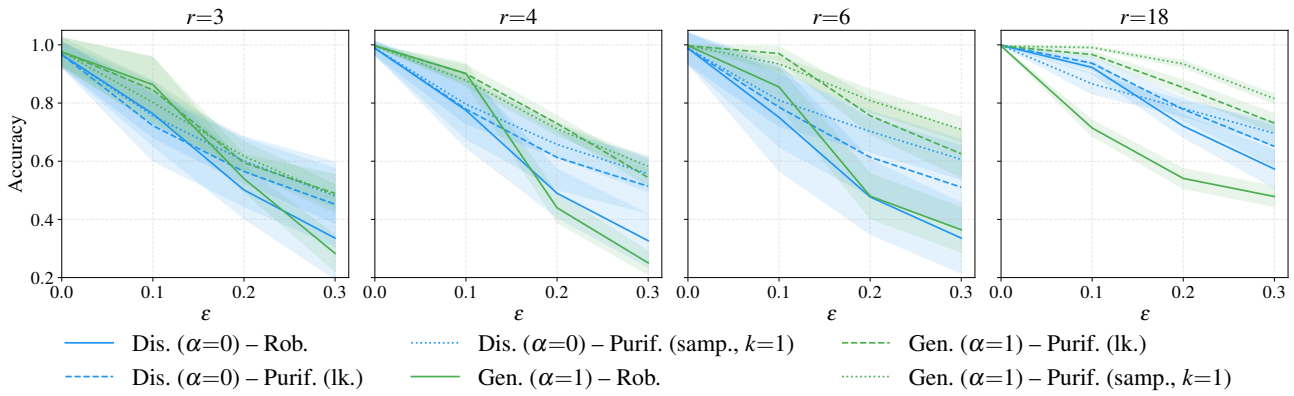


Figure 9: Robustness across attack radii for different model capacities. Each panel fixes the rank r and compares discriminative ($\alpha = 0$) and generative ($\alpha = 1$) models under raw classification, likelihood-based purification, and one-step sampling-based purification. Although the raw robust accuracy of the generative models may decrease for larger capacities, purification yields substantial gains, especially when the density model is sufficiently expressive.

models sit at the low-capacity, underfitting end of this spectrum: the density is too coarse to recover the data manifold, so generative training cannot match adversarial training. The spirals trends suggest that increasing the rank toward the values at which MPS-BMs model such data faithfully would narrow this gap, particularly for purification.

I.2 EFFECT OF FEATURE EMBEDDING

We next compare different orthonormal feature embeddings. Besides the Legendre embedding used in the main experiments, we consider Fourier, Hermite, and Chebyshev embeddings of the first and second kind. The Fourier embedding is defined on $[0, 1]$ by

$$\phi_1(x) = 1, \qquad \phi_{2k}(x) = \sqrt{2}\cos(2\pi kx), \qquad \phi_{2k+1}(x) = \sqrt{2}\sin(2\pi kx). \tag{6}$$

On $[-1, 1]$, the Legendre embedding is given by

$$\phi_k(x) = \sqrt{\frac{2k+1}{2}}\, P_k(x), \qquad k = 0, 1, \ldots, d-1, \tag{7}$$

where P_k is the Legendre polynomial of degree k. We also use Chebyshev embeddings obtained from the corresponding orthogonal polynomials after absorbing the square root of the weight into the feature map:

$$\phi_0^{\mathrm{ChebI}}(x) = \frac{(1-x^2)^{-1/4}}{\sqrt{\pi}}\, T_0(x), \qquad \phi_n^{\mathrm{ChebI}}(x) = \sqrt{\frac{2}{\pi}}\,(1-x^2)^{-1/4}\, T_n(x), \quad n \geq 1. \tag{8}$$

$$\phi_n^{\mathrm{ChebII}}(x) = \sqrt{\frac{2}{\pi}}\,(1-x^2)^{1/4}\, U_n(x), \quad n \geq 0. \tag{9}$$

For Chebyshev I, the domain is restricted to $(-0.99, 0.99)$ to avoid the endpoint divergence. Finally, the Hermite embedding uses the normalized Hermite functions

$$\phi_n^{\mathrm{Herm}}(x) = \frac{H_n(x)e^{-x^2/2}}{\sqrt{\sqrt{\pi}\, 2^n n!}}, \qquad n = 0, 1, \ldots, d-1, \tag{10}$$

with inputs rescaled to lie in the central support of the Gaussian weight.

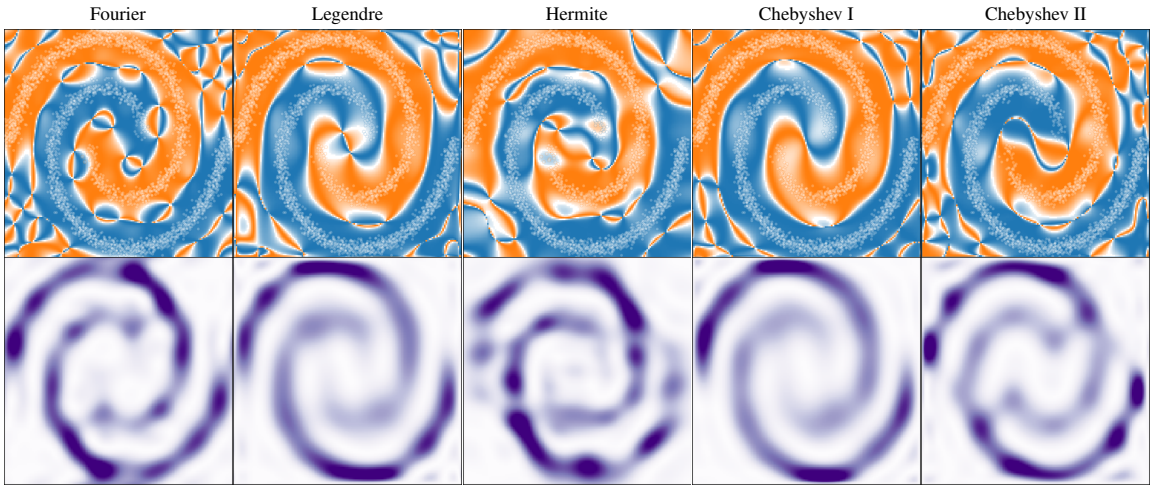


Figure 10: Effect of the feature embedding on the learned classifier and density for generative models with $d = 10$ and $r = 6$. Top row: conditional distribution $p_T(c \mid x)$ as decision regions, with dataset points overlaid in light gray. Bottom row: marginal input density $p_T(x)$. Different embeddings lead to different decision boundaries and density estimates, reflecting their distinct functional biases.

Figure 10 shows that the embedding choice has a visible effect even when d and r are fixed. Different embeddings induce different functional biases, leading to different decision boundaries and density estimates. Thus, the embedding is not merely a preprocessing detail: it helps determine which functions can be efficiently represented by the MPS.

The same variability appears in the quantitative metrics in Figure 11. No embedding is uniformly best across all metrics: some improve raw robust accuracy, while others favor detection or purification. Overall, Legendre provides a balanced choice on this dataset, which motivates its use in the main experiments.

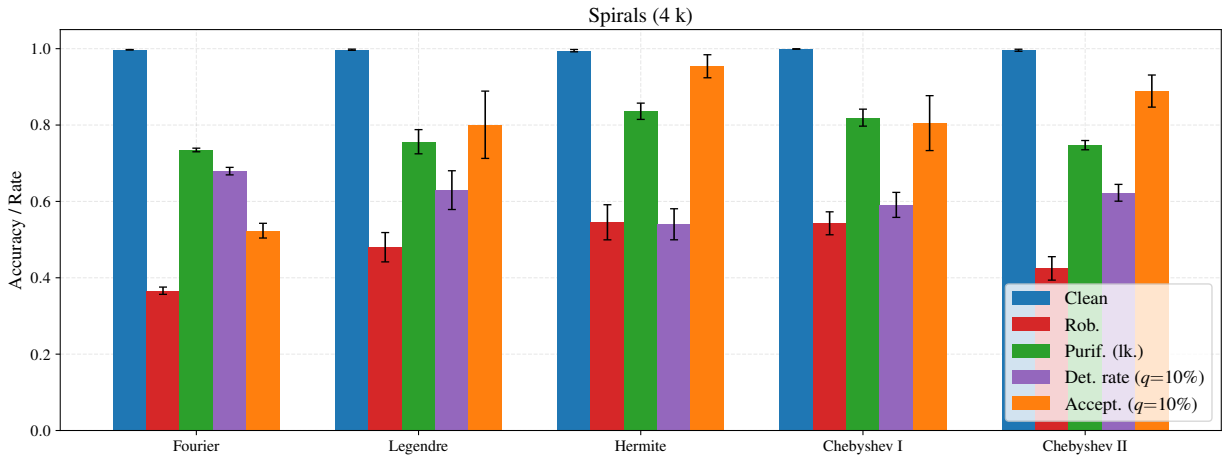


Figure 11: Effect of the feature embedding on robustness metrics for generative models with $d = 10$ and $r = 6$. Bars show clean accuracy, raw robust accuracy, likelihood-purified accuracy, adversarial detection rate, and accuracy on accepted samples after likelihood-based detection with $q = 10\%$. Different embeddings lead to different trade-offs across metrics, with Legendre providing a balanced choice for the main experiments.

J DETECTOR-AWARE JOINT-LIKELIHOOD ATTACK

The attacks in the main text target the classifier through the conditional probability $p_T(c \mid x)$. Since likelihood-based detection and purification instead use the learned joint density, we also evaluate a detector-aware joint-likelihood attack. We stress what this attack is and is not: it is aware of the density the detector thresholds, and so changes the attack *objective*, but it is still computed against the model alone and does not backpropagate through the purifier. It therefore does not answer the purifier-aware question raised in Appendix C. For a clean example (x, c), the attack searches inside the same ℓ_∞ ball for a

point with high joint likelihood under an incorrect class:

$$\tilde{x} \in \operatorname*{arg\,min}_{x' \in B_\varepsilon^{(\infty)}(x)} \min_{c' \neq c} \left[-\log p_T(x', c') \right]. \tag{11}$$

Equivalently, the attack maximizes the wrong-class joint likelihood, aiming to produce adversarial examples that are both misclassified and plausible under the learned density.

Table 3 compares this attack with the conditional attack at $\varepsilon = 0.2$. Values report the difference between the joint-likelihood attack and the conditional attack, so higher values indicate that the joint-likelihood attack is weaker for that metric. Overall, the joint-likelihood attack is less effective, especially for models with a generative component. A likely explanation is that, on this simple dataset, these models learn the joint density well, so maximizing the wrong-class joint likelihood succeeds mainly when the perturbation can move the point close to the opposite class manifold. Under the radius constraint $\varepsilon = 0.2$, this is difficult, making the attack weaker than the conditional one. The lower adversarial detection rate is consistent with the same interpretation: joint-likelihood attacks are less likely to place points in the low-density region between classes, so fewer adversarial examples appear out of distribution to the detector. These results show only that the density-based defenses are not broken by this particular objective, which is weak evidence of robustness: the attack never sees the purifier, and a purifier-aware attacker [Athalye et al., 2018] remains untested (Appendix C). Stronger objectives that combine misclassification with explicit likelihood-threshold constraints, and that differentiate through the defense, are the natural continuation.

Table 3: Detector-aware joint-likelihood attack compared with the conditional attack on spirals at $\varepsilon = 0.2$. Entries report metric differences, joint-likelihood attack minus conditional attack; hence higher values mean that the joint-likelihood attack is weaker. Reported metrics are robust accuracy, likelihood-purified accuracy, adversarial detection rate, and accuracy on accepted samples, with the detection threshold set to the 10th percentile of validation likelihoods. Values are averages over random seeds $\pm$ one standard deviation, using 20 seeds for the interpolated models and 8 seeds for adversarial training.

Metric	Adv. (ε=0.15)	Dis. (α=0)	Mix. (α=0.5)	Gen. (α=1)
Δ Rob.	0.146 ± 0.027	0.079 ± 0.046	0.226 ± 0.032	0.188 ± 0.037
Δ Purif. (lk.)	-0.006 ± 0.024	0.013 ± 0.019	0.039 ± 0.012	0.024 ± 0.028
Δ Det. (q=10%)	-0.071 ± 0.116	-0.144 ± 0.121	-0.216 ± 0.047	-0.087 ± 0.045
Δ Accept. (q=10%)	0.122 ± 0.038	0.050 ± 0.046	0.063 ± 0.043	0.011 ± 0.058

We repeat the same comparison on MNIST in Table 4, with entries again reporting the joint-likelihood attack minus the conditional attack (negative values indicate that the detector-aware joint attack is stronger). The picture differs from the spirals data because the MNIST models underfit: for the purely discriminative and weakly generative models the joint attack is markedly weaker on raw and purified accuracy, but it does consistently lower the detection rate, confirming that joint-likelihood adversarial examples are harder to flag as out of distribution. For the more generative $\alpha = 0.5$ model the joint attack becomes competitive on raw robustness, consistent with its density being too coarse to separate the wrong-class manifold under the radius constraint.

Metric	α=0	α=0.01	α=0.5	AT
Rob.	$+0.545 \pm 0.012$	$+0.185 \pm 0.010$	-0.086 ± 0.009	$+0.042 \pm 0.001$
Purif. (lk.)	$+0.226 \pm 0.007$	$+0.177 \pm 0.006$	$+0.084 \pm 0.003$	$+0.030 \pm 0.001$
Det. rate (q=10%)	-0.287 ± 0.032	-0.275 ± 0.004	-0.194 ± 0.004	-0.158 ± 0.024
Accept. err (q=10%)	-0.538 ± 0.014	-0.194 ± 0.011	$+0.086 \pm 0.010$	-0.042 ± 0.002

Table 4: Difference (joint adaptive minus standard attack) at $\varepsilon = 0.2$, q=10%, MNIST, Legendre d=3, r=20. Negative $\Rightarrow$ the detector-aware joint attack is stronger (lower accuracy / lower detection rate).