Capsa: A Unified Framework for Quantifying Risk in Deep Neural Networks

Sadhana Lolla*, **Iaroslav Elistratov***, **Alejandro Perez, Elaheh Ahmadi, Daniela Rus, Alexander Amini**

Themis AI, Inc. `themisai.io`

Abstract

The deployment of large-scale deep neural networks in safety-critical scenarios requires quantifiably calibrated and reliable measures of trust. Unfortunately, existing algorithms to achieve risk-awareness are complex and adhoc. We present `capsa`, an open-source and flexible framework for unifying these methods and creating risk-aware models. We unify state-of-the-art risk algorithms under the `capsa` framework, propose a composability method for combining different risk estimators together in a single function set, and benchmark on high-dimensional perception tasks. Code is available at: `github.com/themis-ai/capsa`.

1 Introduction

Neural networks (NNs) continue to push the boundaries of modern artificial intelligence (AI) systems across a wide range of complex real-world domains, from robotics and autonomy [8, 19, 15], to healthcare and medical decision making [14, 37]. While their performance in these domains remains unmatched, modern NNs still encounter sudden, unexpected, and inexplicable failures that are often catastrophic – especially in safety-critical environments. These failures are largely due to systemic issues that propagate throughout the entire modern AI lifecycle, from imbalances [20, 10] and noise [5] in data that lead to algorithmic bias [9, 12, 11, 13, 30, 33] to predictive uncertainty [21, 23, 27] that plagues model performance on unseen or out-of-distribution data. In order to realize the widespread adoption of AI in society, NNs must not only identify these potential failure modes, but also effectively use this awareness to obtain unified and calibrated measures of risk and uncertainty. There is thus a critical need for unified systems that can estimate quantitative risk metrics for any NN model, and in turn integrate this awareness back into the learning lifecycle to improve robustness, generalization, and safety.

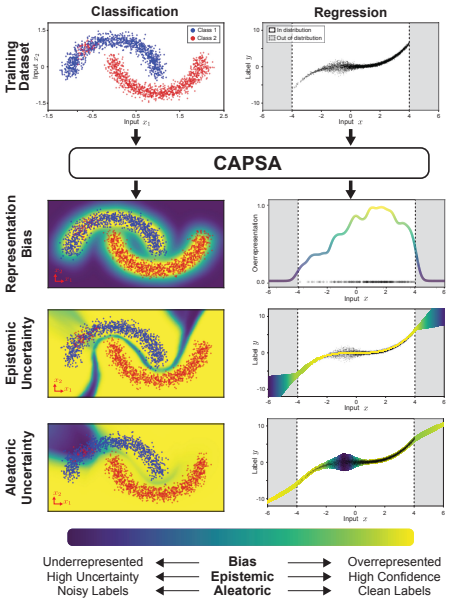

Figure 1: An example of `capsa` wrapping a model to be aware of (top) representation bias, (middle) epistemic uncertainty, and (bottom) aleatoric uncertainty.

To address these fundamental challenges, we present `capsa` – an algorithmic framework for wrapping any arbitrary NN model with state-of-the-art risk-awareness capabilities, shown in 4. By decomposing

*Denotes equal contribution and co-first authorship

the algorithmic stages of risk estimation into their core building blocks, we unify different algorithms and estimation metrics under a common data-centric paradigm. Additionally, because `capsa` renders the underlying NN aware of a variety of risk metrics in parallel, we achieve improved performance and quality in risk estimation. In summary, this paper presents:

1. `Capsa`, an open-source, easy-to-use framework for equipping arbitrary NNs with calibrated awareness of different forms of risk, including bias, label noise, and model uncertainty;
2. An algorithm for decomposing risk estimation algorithms into modular components that can then be composed together to achieve greater efficiency, accuracy, robustness; and
3. Empirical validation of `capsa` on a range of dataset complexities, modalities, applications including algorithmic bias identification, incorrect label discovery, and anomaly detection.

2 Related Work

Existing algorithmic approaches to risk quantification narrowly estimate a singular form of risk in NNs, often in the context of a limited number of data modalities [29, 21, 24, 11, 42]. These methods present critical limitations as a result of their reductionist, ad hoc, and narrow focus on single metrics of risk. However, generalizable methods that provide a holistic awareness of risk have yet to be realized and deployed [27, 39]. This is in part due to the significant engineering changes required to integrate an individual risk algorithm into a larger machine learning system [38, 16, 6, 34], which in turn can impact the quality and reproducibility of results. The lack of a unified algorithm for composing different risk estimation metrics limits the scope and capability of each algorithm independently, and the robustness of the system as a whole.

3 Background and Methodology

3.1 Preliminaries

We consider the problem of supervised learning on a labeled dataset, $\{x, y\}_{i=1}^{n}$. Our goal is to learn a model, f, parameterized by weights, $\boldsymbol{W}$, that minimizes the average loss over our dataset. While the model outputs predictions in the form of $\hat{y} = f_{\boldsymbol{W}}(x)$, we now introduce a risk-aware transformation operation, Φ, which transforms f into a risk-aware variant such that

$$\hat{y}, R = \Phi_{\boldsymbol{\theta}}(f_{\boldsymbol{W}}(x)), \tag{1}$$

where R are the estimated "risk" measures from a set of metrics, $\boldsymbol{\theta}$. The goal of this paper is to propose a common transformation backbone for $\Phi_{\boldsymbol{\theta}}(\cdot)$, which automatically transforms an arbitrary model, f, to be aware of risks, $\boldsymbol{\theta}$. All measures of risk aim to capture, on some level, how trustworthy a given prediction is from a model. We propose the idea of risk *wrappers*, which are instantiations of Φ_θ, for a singular risk metric, θ. Wrappers are given an arbitrary neural network and, while preserving the structure and function of the network, add and modify the relevant components of the model to estimate the risk metric, θ, and still being a drop-in replacement for $f(\cdot)$.

3.2 `Capsa`: The Wrapping Algorithm

We present a unified algorithm building $\Phi_{\boldsymbol{\theta}}$ in order to wrap an arbitrary neural network model. There are four main components: (1) constructing the shared feature extractor, (2) applying modifications to the existing model, (3) creating additional models and augmentations if necessary, and (4) modifying the loss function. We define two types of uncertainty: *data-based* uncertainty, and *model-based* uncertainty (see Appendix A for details) and show that `capsa` can effectively calculate risk metrics associated with both.

The feature extractor, which we define by default as the model until its last layer, can be leveraged as a shared backbone by multiple wrappers at once to predict multiple compositions of risk. This results in a fast, efficient method of reusing the main body of the model. Next, `capsa` modifies the existing network according to metric-specific modifications; this could entail modifying every weight in the model to be drawn from a distribution (to convert to a Bayesian neural network [7]) or adding stochastic dropout layers [17, 3]. Depending on the metric, `capsa` also adds new layers or augmentations to the model that take the feature extractor output and predict new outputs; for

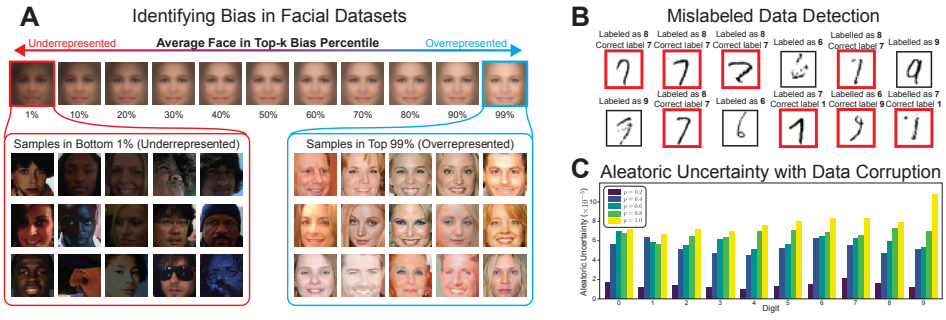

Figure 2: **Bias on faces.** (A) Under-represented and over-represented faces in the Celeb-A dataset found by `capsa` using the VAE and HistogramBias wrappers without cherry-picking. As the percentile bias of the data increases, the skin tone/hair color gets lighter, and lighting gets brighter. **Aleatoric Uncertainty** (B) Artifically mislabeled items in the MNIST dataset have the highest aleatoric uncertainty; and (C) this trend tracks along with the percent of mislabeled examples.

example, new layers to output σ [29, 18], or extra model copies when ensembling [24]. Lastly, we modify the loss function to capture any remaining metric-specific changes that need to be made; for example, KL-divergence [22], negative log-likelihood [29], etc, as shown in 5.

All of the following modifications are integrated together into a custom metric-specific forward pass, and train step to capture variations in the forward and backward passes (shown in 2 of data through the model during training and inference. Our unified wrapping algorithm supports a wide variety of risk estimation methods (Fig. 4) ranging from: (1) **Representational bias** from low-density areas of the feature space(e.g., [22, 32]); (2) **Aleatoric uncertainty** from noisy or incorrect labels [29, 21]; and (3) **Epistemic uncertainty** lack of predictive confidence, measured using Bayesian NNs [7, 17], likelihood estimation [2], ensembling [24], or even reconstruction-based [22] approaches. Details on how `capsa` modifies the input models for all of the above cases are available in the appendix.

3.3 Metric Composability

We propose a novel composability algorithm to create more robust ways of estimating risk (e.g., by combining multiple metrics together into a single metric, or alternatively by capturing different measures of risk independently). We leverage our shared feature extractor as the common backbone of all metrics, and incorporate all model modifications into the feature extractor. New model augmentations are applied either in series or in parallel, depending on the use case. Lastly, the model is jointly optimized using all loss functions by computing the gradient of each loss with regard to the shared backbone's weights and stepping into the direction of the accumulated gradient. Further details are explained in Algorithm 2 and Section B.1.

4 Experiments and Results

In the following section, we analyze the risk metrics obtained by wrapping various models with `capsa` on several datasets. We show that `capsa` provides accurate, scalable, composable risk metrics that are efficient and can be used to quantify bias, aleatoric, and epistemic uncertainty.

Representation Bias – Using `capsa`'s bias wrapper, we analyzed the Celeb-A [26] dataset on the task of facial detection. Fig. 2A qualitatively inspects the different percentiles of bias ranging from underrepresentation (left) to overrepresentation (right). We found that the underrepresented samples in the dataset commonly contained darker skin tones, darker lighting, and faces not looking at the camera. As the percentile of the bias gets higher, we see that the dataset is biased towards lighter skin tones, hair colors, and a more uniform facial direction.

Aleatoric Uncertainty – Next, we test `capsa`'s ability to successfully detects label noise in datasets using aleatoric uncertainty estimation– specifically (MVE) [29]. In the following experiment, we replaced a random collection of the 7s in the MNIST dataset with 8s. As shown in Fig. 2B, the samples with high aleatoric uncertainty are dominated by the mislabeled examples, and also include a naturally mislabeled sample. We further test `capsa`'s sensitivity to mislabeled datasets by artificially corrupting our labels with varying levels of probability p. In Fig. 2C, as p increases, the average

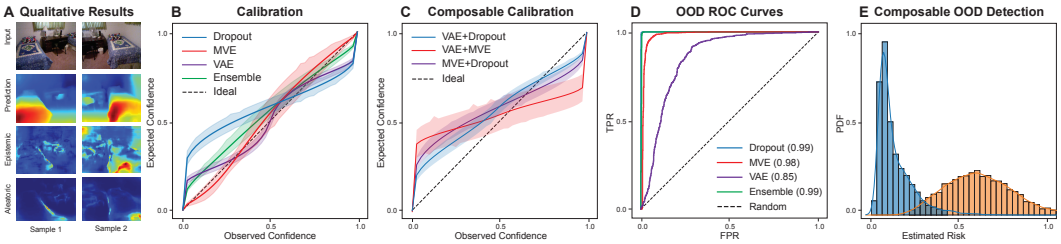

Figure 3: **Risk estimation on monocular depth prediction**. (A) Example pixel-wise depth predictions and uncertainty. Model uncertainty calibration for individual metrics (B) and composed metrics (C). OOD detection assessed via AUC-ROC (D) and a full p.d.f. histogram (E).

uncertainty also increases. These experiments highlight `capsa`'s capability to serve as the backbone of a dataset quality controller and cleaner, due to its high-fidelity aleatoric noise detection.

Epistemic Uncertainty – We demonstrate how `capsa`'s ability to compose multiple methods (e.g., dropout and VAEs) can achieve more robust, efficient performance. We combine aleatoric methods with epistemic methods (i.e., ensembling the MVE metric) to strengthen aleatoric methods (being averaged across multiple runs) or alternatively treat the ensemble of MVEs as a mixture of normals. Similarly a weighted sum of normalized variances compose VAE and dropout.

We demonstrate `capsa` for large-scale risk and uncertainty benchmarking framework for existing methods in the community. To that end, we train a U-Net style model on the task of monocular end-to-end depth estimation (see Tab. 1). Importantly, `capsa` works "out of the box" without requiring any modifications since it is a highly configurable, model-agnostic framework.

Specifically, we use a U-Net style model with a single ouput channel and wrap it with `capsa`. We then train the wrapped model on NYU Depth V2 dataset [28] (27k RGB-to-depth image pairs of indoor scenes) and evaluate on a disjoint test-set of scenes. Additionally, we use outdoor driving images from ApolloScapes [25] as OOD data points.

Table 1: **Monocular depth.** VAE + dropout outperforms all other methods while being more efficient.

	Test Loss	NLL	OOD AUC
Base	0.0027 ± 0.000	–	–
VAE	0.0027 ± 0.000	–	0.8855 ± 0.036
Dropout	0.0027 ± 0.000	0.1397 ± 0.012	0.9986 ± 0.003
Ensembles	**0.0023 ± 0.000**	0.0613 ± 0.022	0.9989 ± 0.002
MVE	0.0036 ± 0.001	**0.0532 ± 0.022**	0.9798 ± 0.012
Dropout + MVE	0.0027 ± 0.000	0.1291 ± 0.015	0.9986 ± 0.003
VAE + Dropout	0.0027 ± 0.000	**0.0932 ± 0.020**	**0.9988 ± 0.002**
VAE + MVE	0.0034 ± 0.001	0.1744 ± 0.016	0.9823 ± 0.010

Another application of `capsa` is for anomaly detection. It is critical for a model to recognize when it is presented with an unreasonable input; in the real world, this could be used for determining when an autonomous vehicle should yield control to a human if the perception system detects that it is presented with such an input. The core idea behind this approach is that a model's epistemic uncertainty on out-of-distribution (OOD) data is higher than on in-distribution (ID) data. Thus, given a risk-aware model we visualize density histograms of per image uncertainty estimates provided by a model on both ID (unseen test-set for NYU Depth V2 dataset) and OOD data (ApolloScapes) (see Fig. 3E). At this point, OOD detection is possible by a simple thresholding set by a validation set. We use AUC-ROC to quantitatively assess the separation of the two density histograms (see Fig. 3D).

5 Conclusions

We present a unified model-agnostic framework for risk estimation, which enables identifying and mitigating safety critical issues in existing models for more trustworthy AI. Our approach opens new avenues for greater reproducibility and benchmarking. We showcase how our method can compose different algorithms together to quantify different risk metrics efficiently in parallel, and be used for the downstream tasks (e.g., bias identification, label cleaning, anomaly detection). We further show how the framework yields interpretable risk estimation results that can provide a deeper insight into decision boundaries of NNs. `Capsa` can further be used to feed model uncertainties back into training processes to debias models [4], perform active learning [41], or quality control [35]. `Capsa` is available at `https://github.com/themis-ai/capsa/` with the goal of accelerating and unifying community advances in the areas of uncertainty estimation and trustworthy AI. In the future, we plan to extend support to other data modalities (e.g., irregular types such as graphs, temporal data, etc), as well as to support other model types.

References

[1] Alexander Amini, Wilko Schwarting, Guy Rosman, Brandon Araki, Sertac Karaman, and Daniela Rus. Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 568–575. IEEE, 2018.

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937, 2020.

[3] Alexander Amini, Ava Soleimany, Sertac Karaman, and Daniela Rus. Spatial uncertainty sampling for end-to-end control. *Advances in Neural Information Processing Systems (NeurIPS) Workshop on Bayesian Deep Learning*, 2018.

[4] Alexander Amini, Ava P Soleimany, Wilko Schwarting, Sangeeta N Bhatia, and Daniela Rus. Uncovering and mitigating algorithmic bias through learned latent structure. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 289–295, 2019.

[5] Eyal Beigman and Beata Beigman Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, 2009.

[6] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[8] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[9] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

[10] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[11] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[12] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[13] Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? *Advances in neural information processing systems*, 31, 2018.

[14] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

[15] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[16] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

[17] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[18] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. In *International Conference on Learning Representations*, 2019.

[19] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 251–257. IEEE, 2020.

[20] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Benjamin Kompa, Jasper Snoek, and Andrew L Beam. Second opinion needed: communicating uncertainty in medical machine learning. *NPJ Digital Medicine*, 4(1):1–6, 2021.

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[25] Miao Liao, Feixiang Lu, Dingfu Zhou, Sibo Zhang, Wei Li, and Ruigang Yang. Dvi: Depth guided video inpainting for autonomous driving. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[27] Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael W Dusenberry, Sebastian Farquhar, Qixuan Feng, Angelos Filos, Marton Havasi, Rodolphe Jenatton, et al. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

[28] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[29] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[30] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

[31] Esther Puyol-Antón, Bram Ruijsink, Stefan K Piechnik, Stefan Neubauer, Steffen E Petersen, Reza Razavi, and Andrew P King. Fairness in cardiac mr image analysis: an investigation of bias due to data imbalance in deep learning based segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 413–423. Springer, 2021.

[32] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pages 832–837, 1956.

[33] Laleh Seyyed-Kalantari, Haoran Zhang, Matthew McDermott, Irene Y Chen, and Marzyeh Ghassemi. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nature medicine*, 27(12):2176–2182, 2021.

[34] Jiaxin Shi, Jianfei Chen, Jun Zhu, Shengyang Sun, Yucen Luo, Yihong Gu, and Yuhao Zhou. Zhusuan: A library for bayesian deep learning. *arXiv preprint arXiv:1709.05870*, 2017.

[35] Ava P Soleimany, Alexander Amini, Samuel Goldman, Daniela Rus, Sangeeta N Bhatia, and Connor W Coley. Evidential deep learning for guided molecular property prediction and discovery. *ACS central science*, 7(8):1356–1367, 2021.

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[37] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019.

[38] Dustin Tran, Alp Kucukelbir, Adji B Dieng, Maja Rudolph, Dawen Liang, and David M Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

[39] Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

[40] Qingyang Xu, Elaheh Ahmadi, Alexander Amini, Daniela Rus, and Andrew W Lo. Identifying and mitigating potential biases in predicting drug approvals. *Drug Safety*, 45(5):521–533, 2022.

[41] Yazhou Yang and Marco Loog. Active learning using uncertainty information. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2646–2651. IEEE, 2016.

[42] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

A Risk Metrics and Background

In this section, we outline three high-level categories of risk which capture the different forms of risk metrics that we quantitatively define and estimate.

Representation Bias - The representation bias of a dataset uncovers imbalance in the feature space of a dataset and captures whether certain combinations of features are more prevalent than others. Note this is fundamentally different from label imbalance, which only captures distributional imbalance in the labels. For example, in driving datasets, it has been demonstrated that the combination of straight roads, sunlight, and no traffic is higher than any other feature combinations, indicating that these samples are overrepresented [1]. Similar has been shown for facial detection [11, 4], medical scans [31], and clinical trials [40]. Uncovering feature representation bias is a computationally expensive process as these features are (1) often unlabeled, and (2) extremely high-dimensional (e.g., images, videos, language, etc), but can be estimated by learning the density distribution of the data. We accomplish this by estimating densities in the feature space. For high-dimensional feature spaces we estimate a low-dimensional embedding using a variational autoencoder [22] or by using the features from the penultimate layer of the model. Bias are the estimated as the imbalance between parts of the density space estimated either discretely (using a discretely-binned histogram) or continuously (using a kernel distribution [32]).

Aleatoric Uncertainty- Aleatoric uncertainty captures noise in the data: mislabeled datapoints, ambiguous labels, classes with low separation, etc. We model aleatoric uncertainty using Mean and Variance Estimation (MVE) [29]. In the regression case, we pass the outputs of the model's feature extractor to another layer that predicts the standard deviation of the output. We train using NLL, and use the predicted variance as an estimate of the aleatoric uncertainty. We apply a modification to the algorithm to generalize also to the classification case in Alg. 1. We assume the classification logits are drawn from a normal distribution and stochastically sample from them using the reparametrization trick. We average stochastic samples and and backpropogate using cross entropy loss through those logits and their inferred uncertainties.

Algorithm 1 Aleatoric Uncertainty in Classification

$\mu, \sigma \leftarrow f_{\boldsymbol{W}}(x)$ ▷ Inference
for $i \in 1..T$ **do** ▷ Stochastic logits
  $\tilde{z} \leftarrow \mu + \sigma \times \epsilon \sim \mathcal{N}(0, 1)$
end for
$\tilde{z} \leftarrow \frac{1}{N} \times \sum_{i=1}^{T} \tilde{z}$ ▷ Average logit
$\hat{y} \leftarrow \frac{\exp(\tilde{z})}{\sum_j \exp(\tilde{z}_j)}$ ▷ Softmax probability
$\mathcal{L}(x, y) \leftarrow -\sum_j y_j \log p_j$ ▷ Cross entropy loss

Epistemic Uncertainty- Epistemic uncertainty measures uncertainty in the model's predictive process – this captures scenarios such as examples that are "hard" to learn, examples whose features are underrepresented, and out-of-distribution data. We provide a unified approach for a variety of epistemic uncertainty methods ranging from Bayesian neural networks [7], ensembling [24], and reconstruction-based [22] approaches. Below, we outline three metrics and how they each fit into `capsa`'s unified risk estimation framework.

A *Bayesian neural network* can be approximated by stochastically sampling, during inference, from a neural network with probabilistic layers [7, 17]. Adding dropout layers [36] to a model is one of the simplest ways to capture epistemic uncertainty [17]. To calculate the uncertainty, we run T forward passes, which is equivalent to Monte Carlo sampling. Computing the first and second moments from the T stochastic samples yields a prediction and uncertainty estimate, respectively.

An *ensemble* of N models, each a randomly initialized stochastic sample, presents a gold-standard approach to accurately estimate epistemic uncertainty [24]. However, this comes with significant computational costs. To reduce the cost of training ensembles, `capsa` automates the construction and management of the training loop for all members of the ensemble and parallelizes their computation.

Variational autoencoders (VAEs) are typically used to learn a robust, low-dimensional representation of the latent space. They can be used as a method of estimating epistemic uncertainty by using the reconstruction loss $MSE(\hat{x}, x)$ - in cases of out-of-distribution data, samples that are hard to learn, or underrepresented samples, we expect that the VAE will have high reconstruction loss, since the mapping to the latent space will be less accurate. Conversely, when the model is very familiar with the features being fed in, or the data is in distribution, we expect the latent space mapping to be robust and the reconstruction loss to be low. To construct the VAE for any given model in `capsa`, we use the feature extractor as the encoder, and reverse the feature extractor automatically when possible to create a decoder.

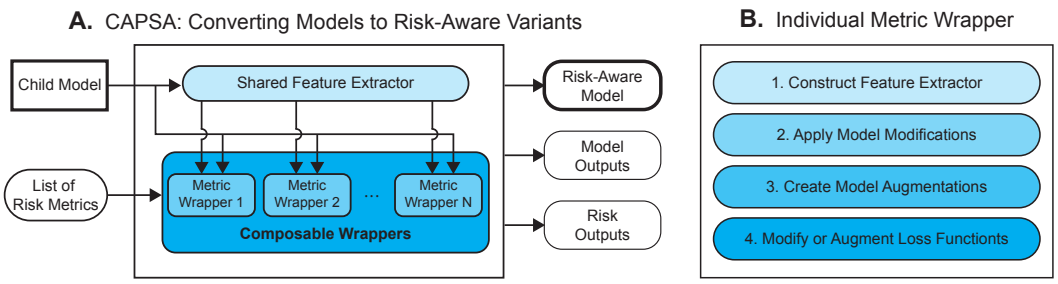

Figure 4: **Overview of `Capsa` architecture**. (A) `Capsa` converts arbitrary NN models into risk-aware variants, that can simultaneously predict both their output along with a list of user-specified risk metrics. (B) Each risk metric forms the basis of a singular model wrapper which is constructed through metric-specific modifications to the model architecture and loss function.

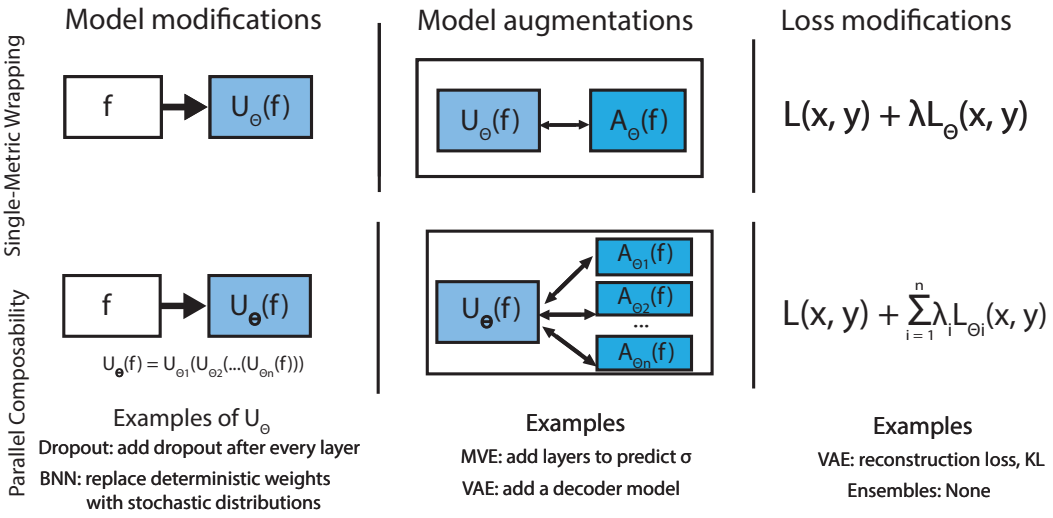

Figure 5: **`Capsa` wrapping algorithm**. (A) Given a specific metric, `capsa` modifies the model, augments it further if necessary, and adds terms to the loss function (B) Applying risk metrics in parallel means that the feature extractor is modified by in series, and then specific layers and loss funcions are independent of each other.

B Additional Methodology

B.1 Composability

We can compose results from multiple uncertainty estimation metrics in two ways: by applying them in parallel or in series. Applying metrics in parallel means that we independently estimate two metrics θ_1 and θ_2, where the only thing shared is the feature extractor (to avoid repeated computation). The modifications to the feature extractor, denoted here as U_θ, are applied in series as we utilize the same feature extractor for all metrics. In order to combine the predictions, we can normalize and sum them. We use this when combining multiple predictions of the same type: i.e. VAEs and Dropout, or Ensembles and VAEs, since the quantity being estimated– model uncertainty– is the same in both cases. Results from this type of composability are shown in 3.

Another method of composability involves combining wrappers in series. This means that we wrap a model first with θ_1, and then subsequently wrap again with θ_2. This results in the layers after the histogram added by θ_1 to be subsequently modified by θ_2. A concrete example of this is ensembling an MVE metric. To wrap a model with MVE, we add in extra layers and change the loss function formulation. When we subsequently wrap this with an EnsembleWrapper with N members, we measure N distributions estimated by the MVE. We can combine these N distributions either by

Algorithm 2 Composed backwards pass

$f, A, \mathcal{L} \leftarrow wrap(m, \Theta)$ ▷ Wrapped user model feature extractor and last layer
$z \leftarrow f(x)$
$\frac{\partial \mathcal{L}}{\partial z} \leftarrow 0$
procedure TRAIN(x, y)
  for $\theta \in \Theta$ **do**
    $\hat{y} \leftarrow A_\theta(z)$ ▷ Metric-customized last layer
    $\ell \leftarrow \mathcal{L}(x, y) + L_\theta(x, y)$
    $\frac{\partial \mathcal{L}}{\partial z} \leftarrow \frac{\partial \mathcal{L}}{\partial z} + \frac{\partial \ell}{\partial z}$ ▷ Accumulate gradients wrt feature extractor
    $A_{\theta_w} \leftarrow A_{\theta_w} - \eta \nabla_{A_\theta}(\ell)$ ▷ Directly update parameters of last layers
  end for
  $\frac{\partial \mathcal{L}}{\partial f_w} \leftarrow \frac{\partial \mathcal{L}}{\partial z} * \frac{\partial z}{\partial f_w}$ ▷ Calculate gradient for feature extractor
  $f_w \leftarrow f_w - \eta \frac{\partial \mathcal{L}}{\partial f_w}$ ▷ Update weights of feature extractor
end procedure

averaging them for a more robust estimate of the MVE, or we can treat the distributions as a mixture of normals as done in [24].

Algorithm 2 shows the computation of the custom backwards pass when combining metrics in parallel. For every metric, we update the corresponding metric-specific augmentations, denoted by A_θ, directly and accumulate the gradients for the feature extractor for every loss function $\mathcal{L}_\theta$. After calculating all of the metrics, we update the weights of the feature extractor accordingly.

C Additional Experiments

C.1 Bias

With our approach, we highlight a critical difference between bias and epistemic estimation methods. The samples estimated to have the highest epistemic uncertainty were not necessarily only underrepresented, but also contain features that obscure the predictive power of the model (e.g., faces with colored lighting, covering masks, and artifacts such as sunglasses and hats), shown in Figure 6.

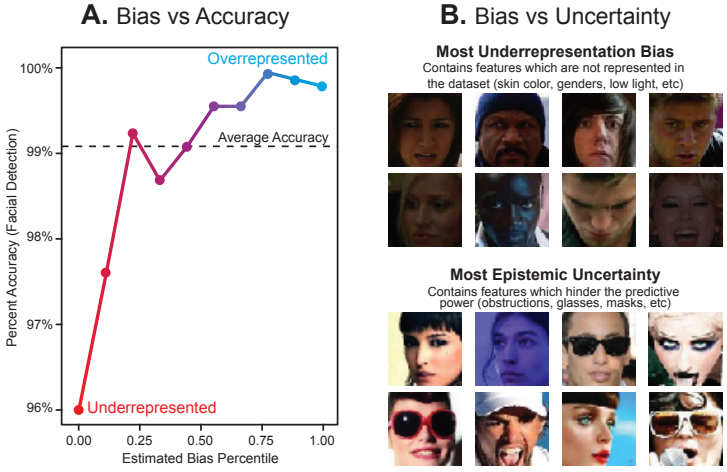

Figure 6: **Epistemic and Biased Samples** (A) Accuracies of samples that the dataset is biased against are lower than those it is biased towards (B) Difference between underrepresented data and epistemic data. Underrepresented samples contain darker skin tones and lighting, while samples with high epistemic uncertainty contain rare artifacts.

Using the bias tools provided by `capsa`, one application is to not only estimate and identify imbalance in the dataset (which we show also leads to performance bias) but to actively reduce the performance bias by adaptively re-sampling datapoints depending on their estimated representation bias during the course of training. The benefits of this are twofold – we can improve sample efficiency by training on less data if some data is redundant, and we can also oversample from areas of the dataset where our latent representation is more sparse.

By composing multiple risk metrics together (in this case, VAEs and histogram bias) we can achieve even greater robustness during training, more sample efficiency, and combine epistemic uncertainty and bias to reduce risk while training.

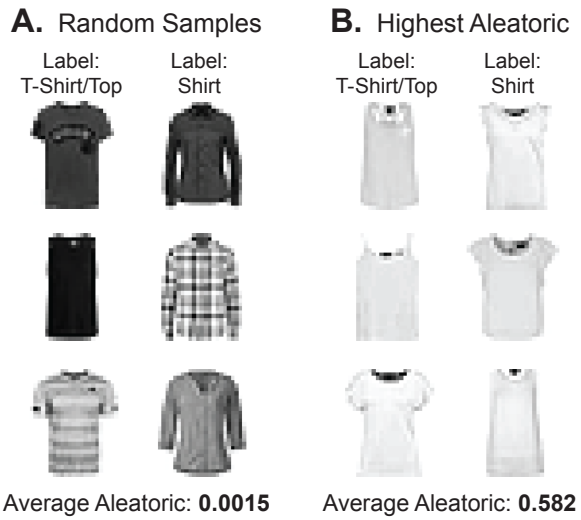

Figure 7: **Fashion MNIST Aleatoric Uncertainty** (A) Randomly selected samples from two classes of fashion-mnist. These samples are visually distinguishable, and have a low aleatoric uncertainty, as opposed to (B), which shows samples with highest estimated aleatoric noise. It is not clear what features distinguish these shirts from tshirts/tops, as they have similar necklines, sleeve lengths, and cuts.

C.2 Aleatoric Uncertainty

In addition to MNIST, we also experiment with natural noise in the FashionMNIST dataset, which contains two very similar classes: "tshirt/top" and "shirt". The methods presented in `capsa` identify samples in Fashion-MNIST with high aleatoric uncertainty in Figure 7, which are light sleeveless tops with similar necklines with minimal visual differences. Short-sleeved shirts with round necklines are also classified as either category. Compared to randomly selected samples from these two classes, the samples that `capsa` marks as noisy are visually indistinguishable, and difficult for humans (and models) to categorize.