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

Anonymous authors
Paper under double-blind review

Abstract

The modern pervasiveness of large-scale deep neural networks (NNs) is driven by their extraordinary performance on complex problems but is also plagued by their sudden, unexpected, and often catastrophic failures, particularly on challenging scenarios. Unfortunately, existing algorithms to achieve risk-awareness of NNs are complex and ad-hoc. Specifically, these methods require significant engineering changes, are often developed only for particular settings, and are not easily composable. Here we present `capsa`, a flexible framework for extending models with risk-awareness. `Capsa` provides principled methodology for quantifying multiple forms of risk and composes different algorithms together to quantify different risk metrics in parallel. We validate `capsa` by implementing state-of-the-art uncertainty estimation algorithms within the `capsa` framework and benchmarking them on complex perception datasets. Furthermore, we demonstrate the ability of `capsa` to easily compose aleatoric uncertainty, epistemic uncertainty, and bias estimation together in a single function set, and show how this integration provides a comprehensive awareness of NN risk.

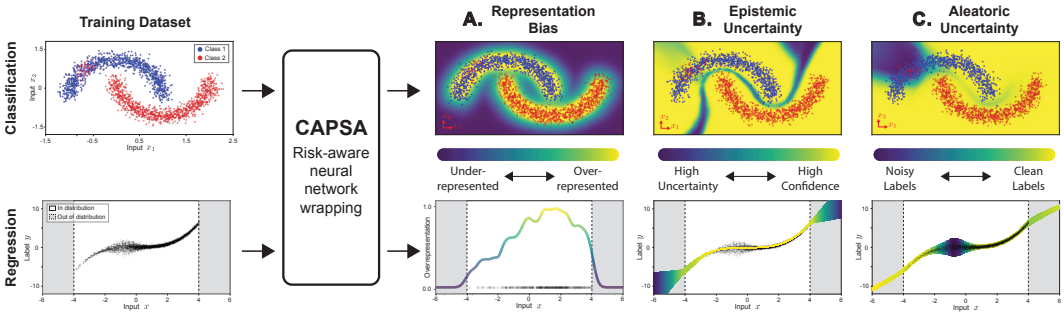

Figure 1: **`Capsa`** unifies state-of-the-art algorithms for quantifying neural network risks ranging from (A) under-representation bias, (B) epistemic (model) uncertainty, and (C) aleatoric uncertainty (label noise). `Capsa` converts existing models into risk-aware variants, capable of identifying risks efficiently during training and deployment.

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 (Bojarski et al., 2016; Hawke et al., 2020; Codevilla et al., 2018), to healthcare and medical decision making (Ching et al., 2018; Topol, 2019). 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 (He & Garcia, 2009; Buda et al., 2018) and noise (Beigman & Klebanov, 2009) in data that lead to algorithmic bias (Bolukbasi et al., 2016; Caliskan et al., 2017; Buolamwini & Gebru, 2018; Chen et al., 2018; Obermeyer

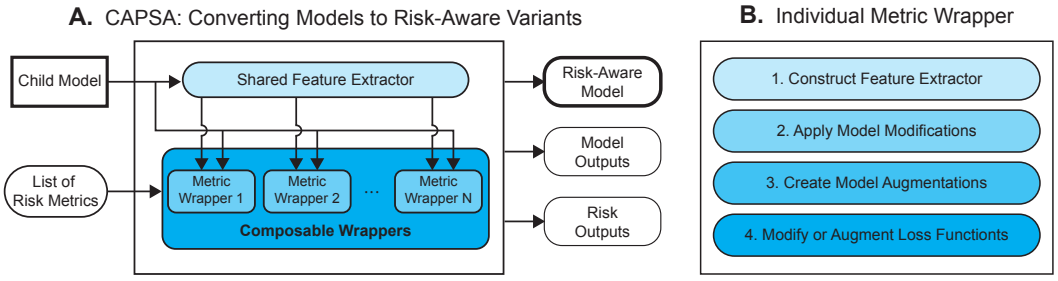

Figure 2: **Overview of `Capsa` architecture**. (A) `Capsa` converts arbitrary NN models into risk-aware variants, that can simultaneously predict both their output along with a list of user-specified risk metrics. (B) Each risk metric forms the basis of a singular model wrapper which is constructed through metric-specific modifications to the model architecture and loss function.

et al., 2019; Seyyed-Kalantari et al., 2021) to predictive uncertainty (Kendall & Gal, 2017; Kompa et al., 2021; Nado et al., 2021) that plagues model performance on unseen or out-of-distribution data. In order to realize the widespread adoption of AI in society, NN models 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.

Existing algorithmic approaches to risk quantification narrowly estimate a singular form of risk in AI models, often in the context of a limited number of data modalities Nix & Weigend (1994); Kendall & Gal (2017); Lakshminarayanan et al. (2017); Buolamwini & Gebru (2018); Zhang et al. (2018). These methods present critical limitations as a result of their reductionist, ad hoc, and narrow focus on single metrics of risk or uncertainty. However, generalizable methods that provide a larger holistic awareness of risk have yet to be realized and deployed Nado et al. (2021); Tran et al. (2022). This is in part due to the significant engineering changes required to integrate an individual risk algorithm into a larger machine learning system (Tran et al., 2016; Dillon et al., 2017; Bingham et al., 2019; Shi et al., 2017), which in turn can impact the quality and reproducibility of results. The lack of a unified algorithm for composing different risk estimation algorithms or risk-aware models together limits the scope and capability of each algorithm independently, and further limits the robustness of the system as a whole. A general, model-agnostic framework for extending NN systems with holistic risk-awareness, covering both uncertainty and bias, would advance the ability and robustness of end-to-end systems.

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. By decomposing 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` allows renders the underlying NN aware of a variety of risk metrics in parallel, we achieve improved performance and quality in risk estimation through principled redundancy, and open the door to achieving a unified composition and hierarchical understanding of NN risk.

In summary, the key contributions of this paper are:

1. `Capsa` an open-source, flexible, and easy-to-use framework for equipping any given neural network with calibrated awareness of different forms of risk – including bias, label noise, and predictive uncertainty;
2. An algorithm for decomposing different types of risk and their estimation methods into modular components that can in turn be integrated and composed together to achieve greater accuracy, robustness, and efficiency; and
3. Empirical validation of `capsa` on a range of dataset complexities and modalities, and the application of `capsa` for mitigation of algorithmic bias, identification of label noise, and detection of anomalies and out-of-distribution data.

2 BACKGROUND AND METHODOLOGY

2.1 PRELIMINARIES

We consider the problem of supervised learning, where we are given a labeled dataset of n input, output pairs, $\{x, y\}_{i=1}^{n}$. Our goal is to learn a model, f, parameterized by weights, $\boldsymbol{W}$, that minimizes the average loss over the entire dataset: $\sum_i \mathcal{L}(f_{\boldsymbol{W}}(x), y)$. While traditionally, the model, a neural network, outputs predictions in the form of $\hat{y} = f_{\boldsymbol{W}}(x)$, we now introduce a risk-aware transformation operation, Φ, which transforms a model, 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. This can stem from the data source itself (aleatoric uncertainty, or representation bias) or the predictive capacity of the model itself (epistemic uncertainty). Within `capsa`, we define various risk metrics as isolating and measuring these sources of risk. We propose the idea of *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 in order to be a drop-in replacement while being able to estimate the risk metric, θ. Wrappers can be further composed using a set of metrics, $\boldsymbol{\theta}$, that are faster and more accurate than individual metrics.

2.2 CAPSA: THE WRAPPING ALGORITHM

While risk estimation algorithms can take a variety of forms and are often developed in ad hoc settings, 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 needed to capture the uncertainty, (3) creating additional models and augmentations if necessary, and (4) modifying the loss functions.

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, rather than training multiple models and risk estimation methods from scratch. Next, `capsa` modifies the existing network according to metric-specific modifications; for example, this could entail modifying every weight in the model to be drawn from a distribution (to convert to a Bayesian neural network (Blundell et al., 2015)) or adding stochastic dropout layers Gal & Ghahramani (2016). Depending on the metric, `capsa` also adds new layers or augmentations to the model that take the feature extractor output and predict new outputs. Note that these are not modifications to the model, but rather augmentations that are only for the given metric; for example, new layers to output σ Nix & Weigend (1994), or extra model copies when ensembling Lakshminarayanan et al. (2017). Lastly, we modify the loss function to capture any remaining metric-specific changes that need to be made. This entails combining the user-specified loss function with the metric-specific loss (KL-divergence Kingma & Welling (2013), negative log-likelihood Nix & Weigend (1994), etc). 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 of data through the model during training and inference.

2.3 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 (Amini et al., 2018). Similar has been shown for facial detection (Buolamwini & Gebru, 2018; Amini et al., 2019), medical scans (Puyol-Antón et al., 2021), and clinical trials (Xu et al., 2022). 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 (Kingma & Welling, 2013) 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 (Rosenblatt, 1956)).

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) (Nix & Weigend, 1994). 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 (Blundell et al., 2015), ensembling (Lakshminarayanan et al., 2017), and reconstruction-based (Kingma & Welling, 2013) 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 (Blundell et al., 2015; Gal & Ghahramani, 2016). Adding dropout layers (Srivastava et al., 2014) to a model is one of the simplest ways to capture epistemic uncertainty Gal & Ghahramani (2016). 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 Lakshminarayanan et al. (2017). 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.

2.4 Metric Composability

Using `capsa`, we compose multiple risk metrics to create more robust ways of estimating risk (e.g., by combining multiple metrics together into a single metric, or alternatively by capturing different

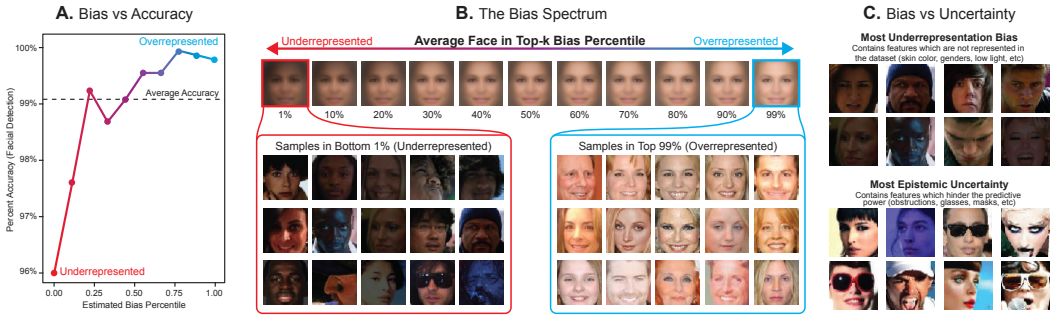

Figure 3: **Bias and Epistemic Uncertainty on Faces** (A) Under-represented and over-represented faces in the Celeb-A dataset found by `capsa` using the VAE and HistogramBias wrappers. As the percentile bias of the data increases, the skin tone gets lighter, lighting gets brighter, and hair color gets lighter, and (B) accuracy on these datapoints increases. We also determine the points with the highest epistemic uncertainty, which have artifacts such as sunglasses, hats, colored lighting, etc.

measures of risk independently). By using the feature extractor as a shared common backbone, we can optimize for multiple objectives, ensemble multiple metrics, and obtain different types of uncertainty estimates at the same time.

We propose a novel composability algorithm within `capsa` to automate and achieve this. Again, we leverage our shared feature extractor as the common backbone of all metrics, and incorporate all model modifications into the feature extractor. Then, we apply the new model augmentations either in series or in parallel, depending on the use case (i.e., we can ensemble a metric in series to average the metric over multiple joint trials, or we can apply ensembling in parallel to estimate a independent measure of risk). Lastly, the model is jointly optimized using all of the relevant 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.

3 Experimental 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 using multiple methods.

3.1 Representation Bias

Using `capsa`'s bias and epistemic wrapper capabilities, we analyzed the Celeb-A Liu et al. (2015) dataset. The task for the neural network was to detect faces from this dataset against non-face images (collated from various negatives in the ImageNet dataset). Fig. 3A quantifies an accuracy vs bias tradeoff that neural networks exhibit, where they tend to perform better on overrepresented training features. We used a VAE as a feature extractor to demonstrate that the VAE can be used for single-shot bias and epistemic uncertainty estimation without any added computation cost.

Fig. 3B 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. With our approach, we highlight a critical difference between bias and epistemic estimation methods in Fig. 3C. 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 mode (e.g., faces with colored lighting, covering masks, and artifacts such as sunglasses and hats).

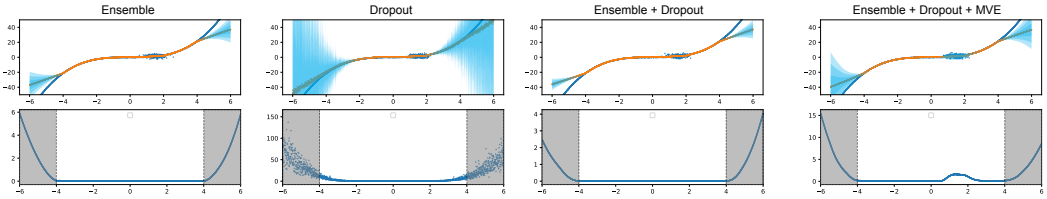

Figure 5: **Risk metrics on cubic regression.** A regression dataset $y = x + \epsilon$, where ϵ is drawn from a Normal centered at $x = 1.5$. Models are trained on $x \in [-4, 4]$ and tested on $x \in [-6, 6]$. Composing using MVE results in a single metric that can seamlessly detect epistemic and aleatoric uncertainty without any modifications to the model construction or training procedure.

3.2 Aleatoric Uncertainty

Next, we experiment on `capsa`'s ability to successfully detects label noise in datasets using aleatoric uncertainty estimation. An example of this can be shown in Fashion-MNIST, which contains two very similar classes: "tshirt/top" and "shirt". The methods presented in `capsa` identify samples in Fashion-MNIST with high aleatoric uncertainty, 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.

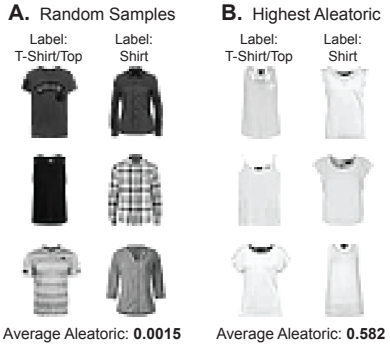

Figure 4: **Fashion MNIST Aleatoric Uncertainty** (A) Randomly selected samples from two classes of fashion-mnist. These samples are visually distinguishable, and have a low aleatoric uncertainty, as opposed to (B), which shows samples with highest estimated aleatoric noise. It is not clear what features distinguish these shirts from tshirts/tops, as they have similar necklines, sleeve lengths, and cuts.

3.3 Epistemic Uncertainty

In this section, we benchmark `capsa`'s epistemic methods on toy datasets. 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, since they are averaged across multiple runs. We can also treat the ensemble of MVEs as a mixture of normals. Similarly, to combine VAE and dropout, we use a weighted sum of their variances or we run the VAE N times with dropout layers and treat the multiple runs as N normals as well.

3.3.1 Cubic Dataset and UCI Benchmarking

We compose various epistemic and aleatoric methods on a cubic dataset with injected aleatoric noise and a lack of data in some parts of the test set. We train models on $y = x+\epsilon$, where $\epsilon \sim \mathcal{N}(1.5, 0.9)$. Training data is within $[-4, 4]$ and test within $[-6, 6]$. Fig. 5 demonstrates that composed metrics can successfully detect the regions with no data, as well as the aleatoric uncertainty in the center.

We also benchmark the raw epistemic uncertainty methods on real-world regression datasets, and evaluate VAEs, ensembles, and dropout uncertainly on these datasets based on Root Mean Squared Error (RMSE) and negative log-likelihood (NLL) in Tab. 1. Additional composability results, as well as training times for all methods, are available in the appendix in Tab. 4 and Tab. 3.

3.3.2 Depth Estimation

In this section, we transition to more complex models and datasets and demonstrate how `capsa` can be used as a 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. 2). Importantly, `capsa` works "out of the box" without requiring any modi-

Table 1: **Regression benchmarking on the UCI datasets**

	RMSE			NLL		
	Dropout	VAE	Ensemble	Dropout	VAE	Ensemble
Boston	2.449 +/- 0.134	2.323 +/- 0.117	2.589 +/- 0.113	2.282 +/- 0.03	2.497 +/- 0.047	2.253 +/- 0.056
Power-Plant	4.327 +/- 0.030	4.286 +/- 0.0120	4.221 +/- 0.028	2.892 +/- 0.00	2.964 +/- 0.00	2.841 +/- 0.005
Yacht	1.540 +/- 0.133	1.418 +/- 0.222	1.393 +/- 0.0965	2.399 +/- 0.03	2.637 +/- 0.131	1.035 +/- 0.116
Concrete	6.628 +/- 0.286	6.382 +/- 0.101	6.456 +/- 0.846	3.427 +/- 0.042	3.361 +/- 0.016	3.139 +/- 0.115
Naval	0.00 +/- 0.000	0.00 +/- 0.000	0.00 +/- 0.00	1.453 +/- 0.667	-2.482 +/- 0.229	-3.542 +/- 0.015
Energy	1.661 +/- 0.090	1.377 +/- 0.091	1.349 +/- 0.175	2.120 +/- 0.022	1.999 +/- 0.113	1.395 +/- 0.066
Kin8nm	0.088 +/- 0.001	0.0826 +/- 0.001	0.072 +/- 0.000	-0.972 +/- 0.01	-0.913 +/- 0.00	-1.26 +/- 0.008
Protein	4.559 +/- 0.031	4.361 +/- 0.0156	4.295 +/- 0.029	4.452 +/- 0.012	3.345 +/- 0.011	2.723 +/- 0.023

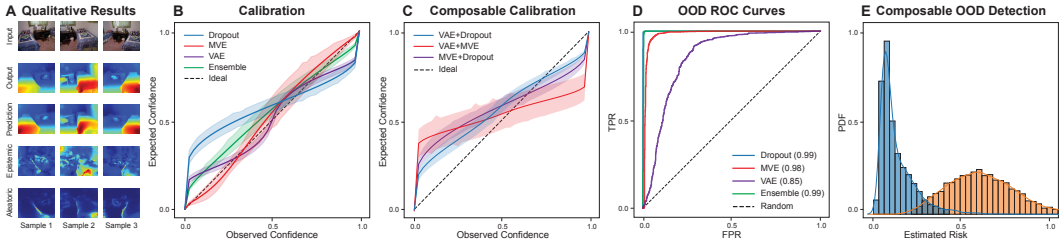

Figure 6: **Risk estimation on monocular depth prediction**. (A) Example pixel-wise depth predictions and uncertainty. Model uncertainty calibration for individual metrics (B) and composed metrics (C). OOD detection assessed via AUC-ROC (D) and a full p.d.f. histogram (E).

fications since it is a highly configurable, model-agnostic framework with modularity as one of the core of its design principles.

Specifically, we take a U-Net style model whose final layer outputs a single $H \times W$ activation map and wrap it with `capsa`. We then train the wrapped model on NYU Depth V2 dataset Nathan Silberman & Fergus (2012) (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 Liao et al. (2020) as OOD data points.

Table 2: **Depth regression results.** VAE + dropout outperforms all other epistemic methods and is more efficient.

	Test Loss	NLL	OOD AUC
Base	0.0027 ± 0.0002	–	–
VAE	0.0027 ± 0.0001	–	0.8855 ± 0.0361
Dropout	0.0027 ± 0.0001	0.1397 ± 0.0123	0.9986 ± 0.0026
Ensembles	**0.0023 ± 7e-05**	0.0613 ± 0.0217	0.9989 ± 0.0018
MVE	0.0036 ± 0.0010	**0.0532 ± 0.0224**	0.9798 ± 0.0118
Dropout + MVE	0.0027 ± 0.0001	0.1291 ± 0.0146	0.9986 ± 0.0026
VAE + Dropout	0.0027 ± 0.0001	**0.0932 ± 0.0201**	**0.9988 ± 0.0024**
VAE + MVE	0.0034 ± 0.0012	0.1744 ± 0.0156	0.9823 ± 0.0102

We see that when we wrap the model with an aleatoric method in Fig. 10, we can successfully detect label noise or mislabeled data – the model exhibits increased aleatoric uncertainty on object boundaries, and indeed we see that the ground truth has noisy labels particularly on the edges of these objects; this could be due to sensor noise or motion noise.

With dropout (Fig. 11) or ensemble (Fig. 12) wrappers, we capture uncertainty in model's prediction itself. We see that increased epistemic uncertainty roughly corresponds to the semantically and visually challenging pixels where the model is making errors.

4 Applications

The benefits of seamlessly and efficiently integrating a variety of risk estimation methods into arbitrary neural models extends far beyond benchmarking and unifying these algorithms within the community. In this section, we outline critically important applications that arise or are aided directly as a result of the estimation abilities we present in `capsa`.

Figure 7: **Debiasing Facial Recognition Systems** (A) Facial datasets are overwhelmingly biased towards light-skinned females. (B) The feature combinations present in dark-skinned males make up only 1.49% of the dataset, and those present in dark-skinned female faces only take up 8.18% of the dataset. Since `capsa` provides us with exactly which datapoints are underrepresented, we can implement a smart sampling scheme to increase the representation of these feature combinations.

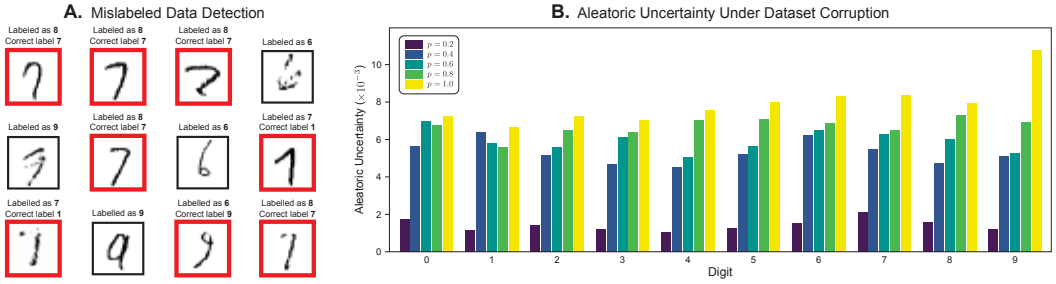

Figure 8: **Mislabeled Examples in the MNIST dataset** (A) If we purposefully inject label noise into the MNIST dataset by labeling 20% of the 7s in the dataset as 8, the mislabeled items have the highest aleatoric uncertainty. We also find a naturally mislabeled sample in the dataset. (B) As the percentage of mislabeled items increases, the average measured aleatoric uncertainty per class also increases.

4.1 Debiasing Facial Recognition Systems

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. As shown in Fig. 7, using `capsa` we pinpoint exactly which samples need under/oversampling, and therefore we can intelligently resample from the dataset during training instead of sampling uniformly. 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.

4.2 Detecting Mislabeled Examples

Another application of `capsa` is cleaning mislabeled or noisy datasets, since we previously described the ability for `capsa` can find noisy labels with high accuracy in Sec. 3.2. In the following experiment, we replaced a random collection of the 7s in the MNIST dataset with 8s. As shown in Fig. 8(A), 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. 8B, as p increases, the average aleatoric uncertainty per class 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.

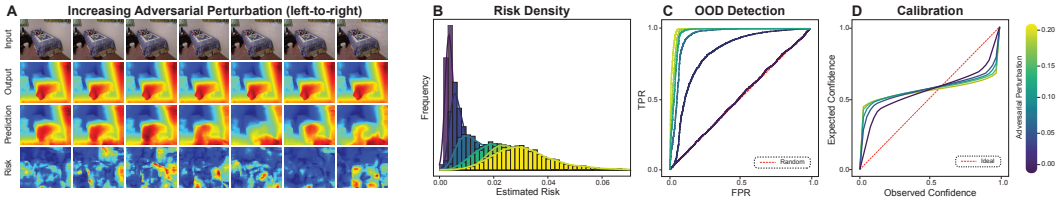

Figure 9: **Robustness under adversarial noise** Across increasing levels of adversarial perturbations: (A) Pixel-wise depth predictions and uncertainty visualizations, (B) Density histograms of per image uncertainty, (C) OOD detection assessed via AUC-ROC, (D) Calibration curves.

4.3 Anomaly and Adversarial Noise Detection

Another application of the "uncertainty estimation" functionality provided by `capsa` is anomaly detection. The core idea behind this approach is that a model's epistemic uncertainty on out-of-distribution (OOD) data is naturally higher than the same model's epistemic uncertainty 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. 6E). At this point, OOD detection is possible by a simple thresholding. We use AUC-ROC to quantitatively assess the separation of the two density histograms, a higher AUC indicates a better quality of the separation (see Fig. 6D) .

It is critical for a model to recognize that it is presented with an unreasonable input (e.g., OOD); in the real world this could be used for an autonomous vehicle yielding control to a human if the perception system detects that it is presented with such an input image as it is expected that model's performance on this datapoint will be poor. For example, in Fig. 9A we see that as the more the image becomes an out of distribution, the more the depth estimate degrades. However, as we are able to detect such a distribution shift, model can pass this information downstream to avoid potentially disastrous prediction before it actually happens.

Further, the approach described above could be used to detect adversarial attacks (perturbations). In Fig. 9A we see that even though the perturbed images are not immediately distinguishable to a human eye, the method described above successfully detects the altered input images.

Another way of interpreting the adversarial perturbations is as a way of gradually turning ID data points into OOD. Such a granular control allows for a better model introspection. In Fig. 9 we see that as the epsilon of the perturbation increases the density histograms of per image uncertainty estimates provided by a model on both the ID and perturbed images become more disentangled (B) and thus the quality of separation increases (C).

5 Conclusions

In this paper, we present a unified model-agnostic framework for risk estimation, which allows for seamless and efficient integration of the uncertainty estimates to the existing models in a couple lines of code. Our approach opens new avenues for greater reproducibility and benchmarking of risk and uncertainty estimation methods across the community. We validate the scalability and the convenience of the framework on a variety of datasets. We showcase how our method can compose different algorithms together to quantify different risk metrics efficiently in parallel, and demonstrate how the obtained uncertainty estimates can be used for the downstream tasks. We further show how the framework yields interpretable risk estimation results that can provide a deeper insight into decision boundaries of the NNs. `Capsa` will be open-sourced as part of this submission 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 our approach to other data modalities including irregular types (graphs) and temporal data (sequences), as well as to support other model types and other risk metrics.

REFERENCES

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)*, pp. 568–575. IEEE, 2018.

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*, pp. 289–295, 2019.

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*, pp. 280–287, 2009.

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.

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

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.

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.

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.

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

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

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

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.

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)*, pp. 4693–4700. IEEE, 2018.

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.

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

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)*, pp. 251–257. IEEE, 2020.

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

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.

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

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.

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

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*, pp. 1–17. Springer, 2020.

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.

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.

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

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, pp. 55–60. IEEE, 1994.

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.

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*, pp. 413–423. Springer, 2021.

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

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.

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.

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.

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

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.

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.

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.

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*, pp. 335–340, 2018.

A APPENDIX

We see that when we wrap the model with an aleatoric method in Fig. 10, we can successfully detect label noise or mislabeled data e.g. in the 4th row on the left we see that the ground truth label has a mislabeled blob of pixels near the right shoulder of a person, the wrapped model is able to detect this and selectively assign high aleatoric uncertainty to this region while leaving the correctly labeled parts of the image "untouched".

	Model Modifications	New Models	Loss Changes	Uncertainty Estimation
Ensemble	None	Train N - 1 additional models	N loss functions and N optimizers	Uncertainty Estimation (Classification)
MVE	Adding layers for sigma	None	Train using NLL for regression, train on perturbed logits for classification	Predicted variance
VAE	None	Adding decoder model	MSE + KL-loss	MSE(x, \hat{x})
Dropout	Dropout layers after fully connected/convolutional layers	None	None	Variance of T forward passes
Histogram Bias	Calculate histogram after every batch	None	None	Joint probability of sample features

Table 3: VAE + Dropout composability Results of composability experiments on UCI datasets. The NLL reduces drastically for most datasets between pure VAE and VAE + Dropout, and the RMSE remains competitive, showing that composability improves uncertainty estimation quality.

	RMSE	**NLL**
Boston	2.278 ± 0.113	2.292 ± 0.056
Power-Plant	4.323 ± 0.017	2.891 ± 0.010
Yacht	1.630 ± 0.241	2.081± 0.072
Concrete	6.489 ± 0.067	3.396 ± 0.036
Naval	0.000 ± 0.000	-2.305 ± 0.144
Energy	1.653 ± 0.200	2.027 ± 0.068
Kin8nm	0.087 ± 0.000	-1.000 ± 0.036
Protein	4.469 ± 0.025	3.156 ± 0.189

Table 4: Training times in seconds of different metrics and composability schemes on real-world regression datasets.

	Ensembles	**Dropout**	**VAE**	**VAE + Dropout**
Boston	11.4 ± 1.1	5.9 ± 1.1	7.8 ± 0.1	7.8 ± 0.2
Power-Plant	80.2 ± 1.2	31.3 ± 0.6	51.1 ± 1.3	54.6 ± 1.0
Yacht	32.4 ± 0.7	14.01 ± 0.1	19.6 ± 0.5	22.1 ± 0.3
Concrete	104.8 ± 2.6	41.4 ± 0.6	57.8 ± 1.5	61.3 ± 0.9
Naval	58.3 ± 0.4	20.0 ± 0.3	33.2 ± 0.5	35.3 ± 0.1
Energy	87.1 ± 0.8	31.7 ± 0.7	43.9 ± 1.3	46.2 ± 1.0
Kin8nm	857.0 ± 57.0	310.6 ± 4.3	440.3 ± 5.8	457.4 ± 11.3
Protein	96.2 ± 0.8	35.7 ± 0.6	59.6 ± 1.0	64.5 ± 1.0

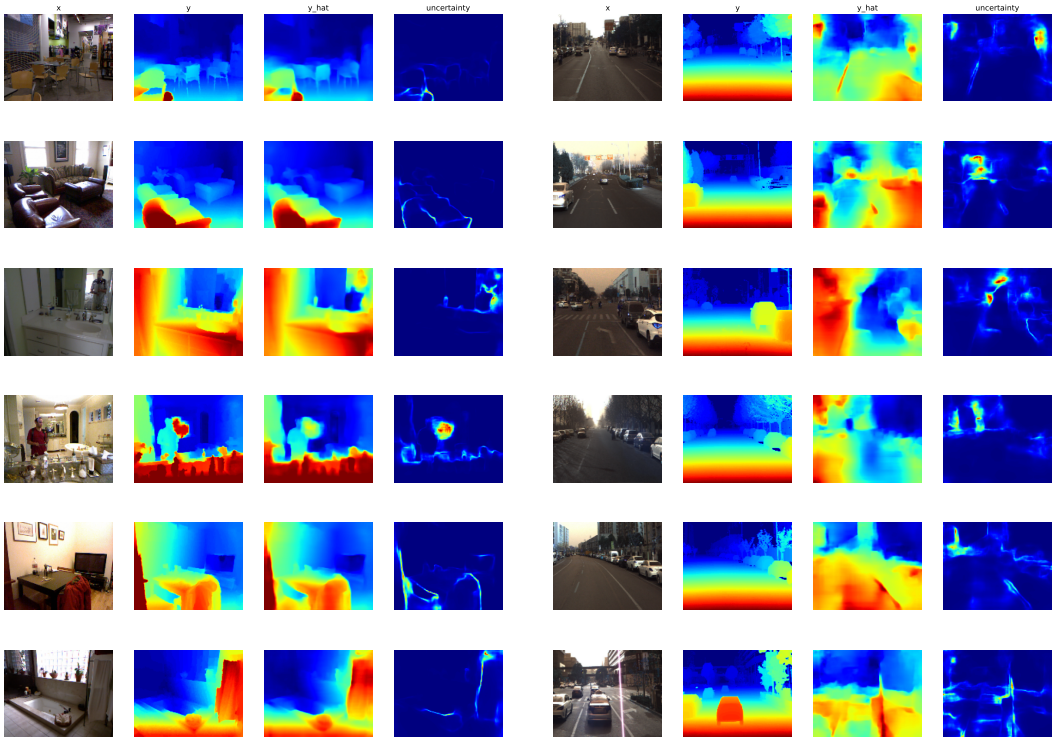

Figure 10: MVE Wrapper

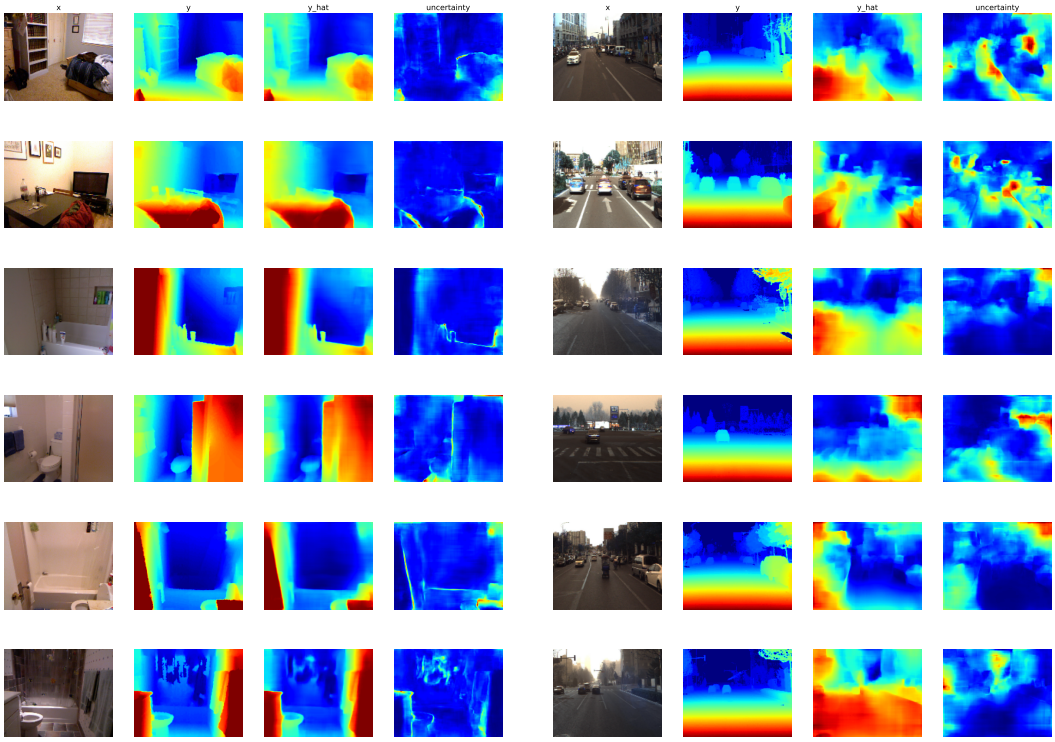

Figure 11: Dropout Wrapper

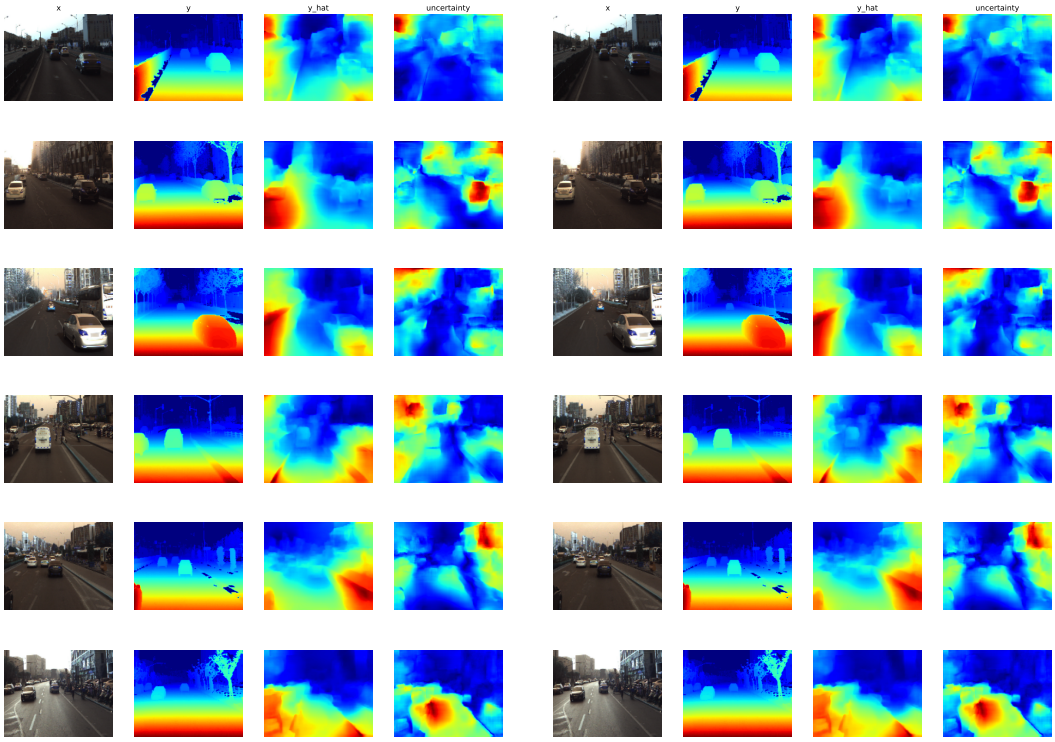

Figure 12: Ensembles Wrapper

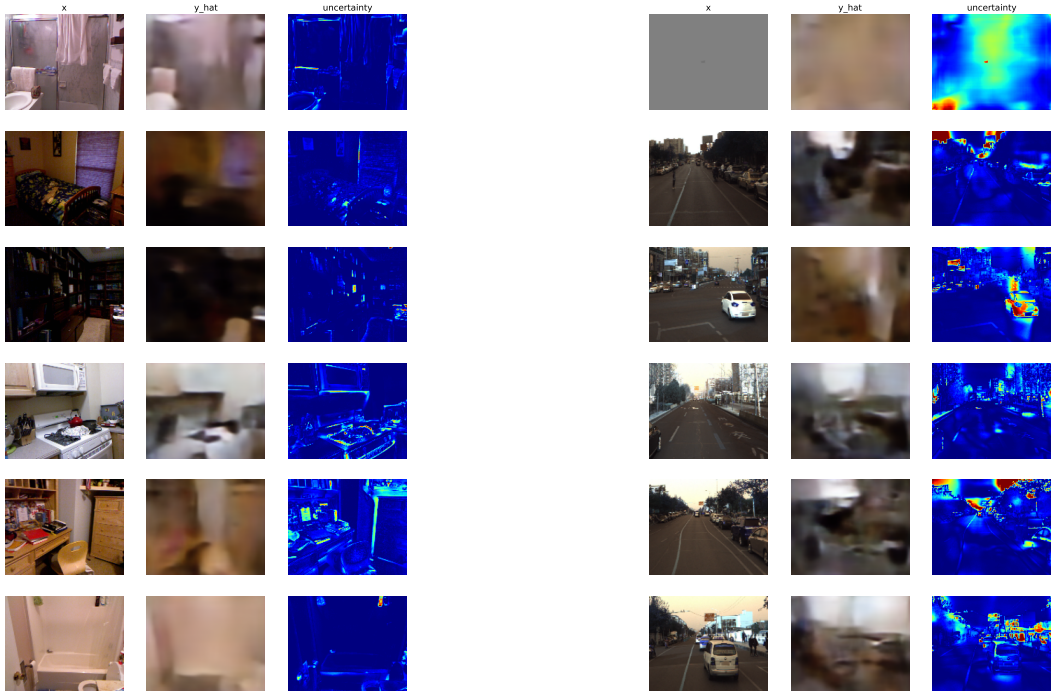

Figure 13: VAE Wrapper