Understanding Uncertainty Maps in Vision with Statistical Testing

Jurijs Nazarovs[1] nazarovs@wisc.edu
Zhichun Huang[2] zhichunh@cs.cmu.edu
Songwong Tasneeyapant[1] tasneeyapant@wisc.edu
Rudrasis Chakraborty[3] rudrasischa@gmail.com
Vikas Singh[1] vsingh@biostat.wisc.edu
[1]University of Wisconsin-Madison [2]Carnegie Mellon University [3]Butlr
https://github.com/vsingh-group/uncertainty_with_rf

Abstract

Quantitative descriptions of confidence intervals and uncertainties of the predictions of a model are needed in many applications in vision and machine learning. Mechanisms that enable this for deep neural network (DNN) models are slowly becoming available, and occasionally, being integrated within production systems. But the literature is sparse in terms of how to perform statistical tests with the uncertainties produced by these overparameterized models. For two models with a similar accuracy profile, is the former model's uncertainty behavior better in a statistically significant sense compared to the second model? For high resolution images, performing hypothesis tests to generate meaningful actionable information (say, at a user specified significance level $\alpha = 0.05$) is difficult but needed in both mission critical settings and elsewhere. In this paper, specifically for uncertainties defined on images, we show how revisiting results from Random Field theory (RFT) when paired with DNN tools (to get around computational hurdles) leads to efficient frameworks that can provide a hypothesis test capabilities, not otherwise available, for uncertainty maps from models used in many vision tasks. We show via many different experiments the viability of this framework.

1. Introduction

With the adoption of deep neural network models in production systems for vision tasks, there is a growing consensus that we must be aware of what our model *does not* know. This is relevant not only for systems used for autonomous driving or medical imaging but also in less critical situations where such a model informs decision making in general and/or is responsible for generating triggers for user intervention. For example, inaccurate but overconfident predictions can lead to undesirable outcomes in assembly line manufacturing and logistics. This need has led to interest in the design of mechanisms for model calibration as well as for estimating uncertainties from deep neural network (DNN) models used in vision for tasks including but not limited to prediction [38, 45], segmentation [3, 41, 47], depth estimation [14, 20] and visual odometry [4, 32].

Figure 1. *Top* shows a raw uncertainty of the depth estimation process. *Bottom* shows **significant** regions selected by our method, with guarantees to restrict family-wise error rate. This region can be used for calibration, model comparisons, or other use cases.

Uncertainties can be roughly categorized into *aleatoric* (statistical) and *epistemic* (systematic). Aleatoric uncertainty can help capture inherent and irreducible data noise, which cannot be reduced even if more data were collected. It can be represented by heteroscedastic models [26, 40], since they assume that the observation noise (uncertainty) can vary with the input. Epistemic uncertainty accounts for uncertainty in model parameters, and can be improved by observing more data. Capturing epistemic uncertainty in a DNN can involve putting a prior on the latent space (e.g., Variational Auto Encoder (VAE) [46]) or model parameters (e.g., Bayesian Neural Networks (BNN) [7, 33, 42]), and adopting any available scheme to estimate the posterior probability. Several strategies exist which use hybrid approaches to capture either aleatoric or epistemic (or both)

by combining heteroscedastic NNs and BNNs, e.g., [30].

Example scenarios. While capturing different types of uncertainties is useful, in practical scientific/industrial settings, uncertainty estimates are merely a "means to an end". We must understand what actions the estimates enable, regardless of whether it is aleatoric or epistemic.

Scenario 1. Uncertainty estimates enable calibration, e.g., by a practitioner evaluating medical images. If a specialist can see that model is uncertain in some specific regions, he/she can evaluate whether to acquire more data if the regions where the model is uncertain are anatomically important. In other cases, such information can guide whether to request a biopsy. However, to decide, we need a statistically sound scheme to generate "significant" uncertain regions. Otherwise, interpreting the raw uncertainty is entirely subjective. Similar applications appear in depth estimation for autonomous vehicles [27], Fig. 1.

Scenario 2. Uncertainty can be used to compare confidences of models. Say a user is satisfied by the accuracy profiles of two models Model_A and Model_B but the second one has a higher latency. An upgrade to Model_B is only justified if one is 99% confident that it reduces uncertainty in a statistically significant sense on a held-out test dataset. This needs a "go/no-go" answer. Similarly, consider two systems for tumor volume dynamics using segmentation, which will drive treatment options (e.g., RECIST criteria [13]). Both systems offer similar accuracy and are FDA approved, but one is more expensive. The investment may be justified if the reduction in uncertainty is significant at a 99.9% level. Alternatively, consider a model on a small form factor device. The choice is between low-precision and high precision operations, the latter will need a larger battery. If both models satisfy client accuracy needs, is the reduction in uncertainty of predictions statistically significant?

Despite the growing body of work on uncertainty, frameworks that enable *actionable* information are limited. The goal of this work is to close this gap.

Classical techniques from statistics. The problems above can be tackled with classical statistical testing. Here, we can set this up as pixel-wise statistical tests (although not strictly necessary; we will discuss alternative forms shortly). Scenario 1 will be a one sample test, while Scenario 2 will be a two sample test: we ask whether the uncertainty at a pixel is different across the two models.

Bottleneck. Deriving a scientifically valid conclusion for the image based on pixel-wise statistical tests will require conducting many tests, equal to the number of pixels. For example, an image of size 28×28 leads to 784 tests. For a common 0.05 critical value (probability of Type-1 error), we expect to select 40 ($\approx 784 \times 0.05$) pixels as significant, purely by chance (number of false positives). This issue escalates for higher resolution images, say 3D medical images. To control a family-wise error rate and avoid inflating the number of false positives, a multiple testing correction (e.g., Bonferroni, Benjamini-Hochberg) [52] is used. However, for high-resolution images common in vision, this tends to over-correct making *none* of the tests significant [2, 56], making the analysis less meaningful.

Many testing setups conservatively assume that the pixels are independent. The classical strategy to *avoid* this restrictive assumption leverages Random Field Theory (RFT), as studied in seminal papers by Adler and Worsley [1,2,57]. However, many theoretical results based on RFT remain restricted to the Gaussian Random Fields (GRF) and some specific generalizations. It is not obvious to what extent these assumptions are viable for uncertainty maps obtained from deep neural networks popular in vision.

Contributions. We show how existing DNN tools when instantiated with suitable results from Random Field theory provide a mechanism to perform hypothesis tests on uncertainty maps, generated by different probabilistic DNN models common in vision. Specifically, we develop a probabilistic framework, based on Neural ODE and Wasserstein distance, which enables learning a diffeomorphism between uncertainty maps and GRFs. We refer to it as *Warping Neural ODE*. Roughly, this allows performing hypothesis tests on the resultant GRFs and mapping results back to the domain of uncertainty maps.

2. Background

In this section, we review several concepts we will use throughout the paper, starting with hypothesis tests.

Hypothesis test is a statistical procedure, which consists of four main parts: **(1)** Null hypothesis H_0 and an Alternative hypothesis H_A, **(2)** test statistics F, **(3)** critical value α, which controls the *probability of Type-1 error*, i.e., $\mathbb{P}\left(\text{reject } H_0 | H_0 \text{ is true}\right) \leq \alpha$ and **(4)** a threshold value $u := u(\alpha)$, which defines the rejection region. While a test statistics, hypotheses and critical value are design choices, the threshold u has to be derived such that the ***p*-value** $\mathbb{P}\left(F \geq u | H_0\right) = \alpha$.

Via hypothesis tests, we can assess whether there is an evidence to reject the null H_0 at a certain level of confidence α. Usually, H_0 states that there is no difference (say, from zero or between two groups), while H_A states that there is a difference. The decision is based on checking whether the observed test statistics F^{obs}, falls in the rejection region, defined by the threshold u.

Family wise error rate (FWER). Recall that the rejection region is selected based on α, which controls $\mathbb{P}\left(\text{Type-1 error}\right)$ of a single test, i.e., $\mathbb{P}\left(\text{reject } H_0 | H_0\right) \leq \alpha$. However, assume that we conduct $N = 100$ tests, e.g., the same test for different pixels, with $\alpha = 0.05$. Then the $\mathbb{P}\left(\text{reject at least one } H_0 | H_0 \text{ is true}\right) = 1 - (1 - \alpha)^N = 0.994$, and on average 5 tests would be rejected purely by chance. For this reason, in multiple comparison testing,

we do not want to control $\mathbb{P}$(Type-1 error), but *FWER:* $\mathbb{P}$(reject at least one $H_0|H_0$). Note that the FWER for 1 test equals to $\mathbb{P}$(Type-1 error).

Gaussian Random Field (GRF). GRF is a family of functions $Z : S \to R$, where for all finite $k \geq 1$ and $\{s_1, \ldots, s_k\} \subset S$, the collection of random variables $\{Z(s_1), \ldots, Z(s_k)\}$ has a multivariate Gaussian distribution. GRFs are parameterized by a mean function $\mu(s) = \mathbb{E}\{f(s)\}$ and covariance function $C(s,t) = \mathbb{E}\{(f(s) - \mu(s))(f(t) - \mu(t))\}$. A *UGRF* is a special Gaussian RF with mean zero, variance 1, and $Var\left(\dot{Z}(s)\right) = I$. An *Isotropic GRF* is a special case where the covariance function $C(s,t)$ depends only on the Euclidean distances $\|s - t\|_2$.

Gaussian related RFs [2] or GRRF is another broad class of random fields $F = f(Z)$, obtained as *functions* of GRF. For example, a Chi-squared RF with d degrees of freedom $\chi^2_d(t) = \sum_{j=1}^{d} Z_j^2(t)$ is a GRRF.

3. Tests on uncertainty maps in vision

We start with an input image $\mathbf{x}$ and address Scenario 1 from §1. Leaving the specific task (depth estimation, segmentation) aside for the moment, we assume that a trained probabilistic model M provides an *uncertainty map* on the input $\mathbf{x}$, denoted as $M_\mathbf{x}$ where $M_\mathbf{x}(s)$ is the uncertainty for pixel $s \in S$. Our model will operate on these uncertainty maps/images (and not on $\mathbf{x}$). To infer which pixels of the uncertainty map (if any) are significant (rather significantly different from 0), we must conduct a hypothesis test.

A standard approach is a test for each pixel s, with $H_0 : M_\mathbf{x}(s) = 0$ and $H_A : M_\mathbf{x}(s) \neq 0$. The test statistics $F(s)$, for example, can be the student statistics

$$F(s) = (\overline{M_\mathbf{x}(s)})/\sigma(\overline{M_\mathbf{x}(s)}),$$

where σ is an estimate of standard deviation and $\overline{(\cdot)}$ represents the sample mean. *For Scenario 2 in §1, to check if there is a difference between uncertainties of models A and B, we replace $M_\mathbf{x}(s)$ by uncertainty $A_\mathbf{x}(s) - B_\mathbf{x}(s)$.*

In the next subsection, we describe how to address the multiple comparison issue. More specifically, we want a procedure that derives a threshold u of the rejection region $\{F(s) \geq u\}$ such that it **(a)** controls **FWER** and **(b)** accounts for spatial correlation of the image.

3.1. Random Field theory to the rescue

We consider the pixel-wise (or voxel-wise for 3D volumes) uncertainty map $M_\mathbf{x}$ as an RF over S with covariance C. Note that the pixel-wise uncertainties may not be independent from each other. We would like to statistically evaluate whether F is different from 0 or not. We will eventually use this strategy to find pixels with significant uncertainty (Scenario 1) and check the difference between uncertainties from two models (Scenario 2). This leads to the hypothesis setup, denoted as $\mathbb{H}_F$:

$$\left[\begin{array}{l} H_0 : \forall s \in S, M_\mathbf{x}(s) = 0 \\ H_A : \exists s \in S, M_\mathbf{x}(s) \neq 0 \end{array}\right. \tag{1}$$

To perform the hypothesis test, it is necessary to establish a test statistic, which ideally describes the nature of the data and is a good indicator of whether to reject the null hypothesis. For RFs, a common test statistic for $\mathbb{H}_F$ is $F_{\max} = \max_{s \in S} M_\mathbf{x}(s)$ [57]. Finally, for the test $\mathbb{H}_F$, we need to find the threshold u_F, such that $\mathbb{P}(F_{\max} \geq u_F|H_0) = \alpha$. Then, if the observed statistics[1] $F^{\text{obs}}_{\max} > u_F$, we may reject H_0 in favor of H_A. However, computing $\mathbb{P}(F_{\max} \geq u_F|H_0)$ is often nontrivial.

Typically, to obtain $\mathbb{P}(F_{\max} \geq u_F|H_0)$, we need to know the theoretical distribution of the test statistics, denoted as $\mathbf{P}_{F_{\max}}$, which may not have a closed form in general. Nevertheless, RF theory provides a way to estimate $\mathbb{P}(F_{\max} \geq u_F|H_0)$ *indirectly*, namely through the *Euler Characteristic Heuristic* (ECH) [53], one of the most important (and fascinating) results in RF theory. Given a u, we define an excursion set $A_u = \{s \in S : F(s) \geq u\}$. ECH shows that, for sufficiently large values u, $\mathbb{P}(F_{\max} \geq u_F|H_0) \approx \mathbb{E}\{\phi(A_{u_F})\}$, where $\phi(A_u)$ is the Euler Characteristic (EC), a well studied quantity in topology that describes the shape of a topological space. Note that hereafter, EEC stands for $\mathbb{E}\{\phi(A_{u_F})\}$.

How to compute $\mathbb{E}\{\phi(A_{u_F})\}$? A standard approach to computing $\mathbb{E}\{\phi(A_{u_F})\}$ is to use Monte Carlo (MC) approximation given Empirical ECs $\widehat{\phi}(A^{\text{obs}}_{u_F})$ over observed excursion sets $A^{\text{obs}}_{u_F}$. However, in addition to the MC approximation error, [1] shows that Empirical ECs at very high levels of u_F are generally too noisy to directly estimate the threshold u_F, such that $\mathbb{P}(F_{\max} \geq u_F|H_0) = 0.05$. In practical setups, it might lead to incorrect hypothesis tests. An alternative approach is to derive the theoretical closed form for $\mathbb{E}\{\phi(A_{u_F})\}$, based on Thm. 3.1 below.

Theorem 3.1 (GKF: Gaussian Kinematic Formula [54])**.** *If F is GRRF (isotropic or non-isotropic), EEC is given as,*

$$\mathbb{P}(F_{\max} \geq u|H_0) \approx \mathbb{E}\{\phi(A_u)\} = \sum_{d=0}^{D} L_d(S, \Lambda(S))\rho_d(u), \tag{2}$$

where D is the dimension of domain S, $\rho_d(u_F)$ is the Euclidean density (ED), $L_d(S, \Lambda(S))$ is the Lipschitz-Killing curvatures (LKC) [58], and $\Lambda(s) = Var(\dot{Z}(s))$ is the variance of the spatial derivative of the underlying UGRF $Z(s)$.

The problem in using (2): Even though (2) applies to a wide range of RFs, it is limited to the availability of the

[1] meaning the statistics $F_{\max}$ for an observed uncertainty map $M^{\text{obs}}_\mathbf{x}(s)$

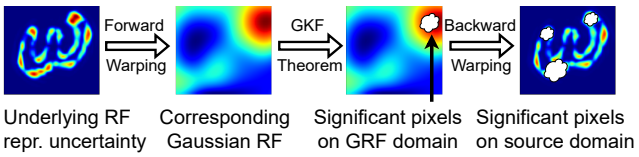

Figure 2. To understand the significant region of the uncertainty map we learn the diffeomorphism warping from general non-isotropic RF to the isotropic GRF. Then, given resulted GRF, we apply Thm. 3.1 to determine the significant region, and warp it back to the source domain.

corresponding $L_d(S)$ and $\rho_d(u_F)$. For most RFs, these are unknown. **(a)** While curvatures $L_d(S)$ are available for isotropic RFs, when dealing with *non-isotropic* RFs, it is often extremely difficult to evaluate $L_d(S)$ [1]. **(b)** Concurrently, the closed form solution for $\rho_d(u_F)$ is available only for a few distributions [8].

Observe that the statistics $F(s)$, representing pixel-wise uncertainty, can in fact be considered as GRRF in several situations. This is because it is a function of Gaussian latent space in VAEs or the weights in BNNs. If so, we can apply Thm. 3.1. However, **(a)** the assumption of a RF F being isotropic on a domain S is unrealistic, which makes closed form solutions $L_d(S)$ defined for isotropic RF inapplicable. **(b)** The exact distribution of F is unknown, and thus ρ_d are also unknown. Then, is there a way to apply Thm. 3.1 on the observed uncertainty maps, generated by a DNN?

3.2. Let us warp to GRFs!

For the development of our proposal, we first (informally) state the following simple result, which describes how the warping of the domain (coordinate system) can help, proofs are in the supplement.

Theorem 3.2. *The domain S of the GRRF F can be warped via a one-to-one smooth transformation Γ to a domain S' without fundamentally changing the problem, namely:* $\mathbb{P}\left(\max_{s'\in S'} F(s') \geq t\right) = \mathbb{P}\left(\max_{s\in S} F(s) \geq t\right)$.

Theorem 3.3. *Consider the GRRF $F(S)$ on the domain S with Euler densities $\{\rho_d^F(u)\}$, and the GRF $Z(S^Z)$ on the domain S^Z with Euler densities $\{\rho_d^Z(u)\}$. Assume that both Euler densities $\{\rho_d^F(u)\}$ and $\{\rho_d^Z(u)\}$ are defined on the same domain $u \in U$ and $\max_d\{\rho_d^F(u)/\rho_d^Z(u)\} \leq 1$. Then, by finding a one-to-one transformation Γ, such that $S = \Gamma S^Z$ and $S^Z = \Gamma^{-1}S$, and selecting a threshold u^*, such that $\mathbb{P}\left(\max_{s\in S^z} Z(s) \geq u^*\right) = 0.05$, guarantees that $\mathbb{P}\left(\max_{s\in S} F(s) \geq u^*\right) \leq 0.05$.*

Remark 1. *For the* **isotropic** *GRF $Z(s)$, all components of Thm. (3.1), L_d and ρ_d are known in a closed form and thus, the corresponding threshold u can be computed:* $\mathbb{P}\left(F_{\max} \geq u | H_0\right) \leq 0.05$.

Based on Thm. 3.3, we could warp the uncertainty map to the isotropic GRF (Fig. 2, 1st arrow). Then based on Remark 1, we apply Thm. 3.1 and derive the significant region (Fig. 2, 2nd arrow), and warp the significant region back (Fig. 2, 3rd arrow). This approach is like [55], which warps a domain of non-isotropic GRF to achieve local isotropy. But in contrast to [55], we seek to find a warping of non-isotropic GRRF to isotropic GRF. To achieve this, we should satisfy **two properties**: **(a)** the learned warping has to be a diffeomorphism, **(b)** the warped version of GRRF should be an isotropic GRF. *That is*, given a general (isotropic or non-isotropic) GRRF $F(S)$ in the source domain S and the isotropic GRF $Z(S^Z)$ on the GRF domain S^Z, we must find a transformation (warping) $\Phi(S)$, such that $F(\Phi(S)) \sim Z(S^Z)$, i.e., equal in distribution. Here, we can use recent developments in machine learning.

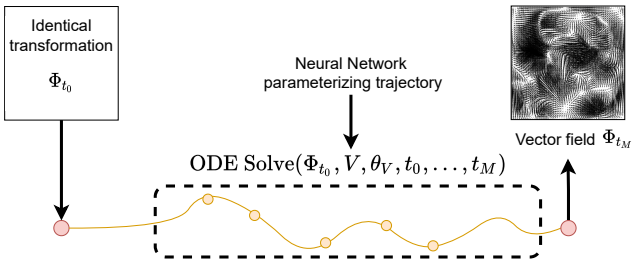

Figure 3. Neural Warping ODE: we model diffeomorphism Φ as a solution of ODE (3), where the RHS is modelled by NN. The resulted transformation Φ_{t_M} is applied to the coordinate system of input image to generate warped domain.

3.2.1 Learning diffeomorphisms

Learning the warp $\Phi(S)$ as a diffeomorphism guarantees invertibility of the transformations, which conserves topological features [49]. For us, this means that we can recover significant regions from the Gaussian domain S^Z, but back in the source domain S, Fig. 2 (3rd arrow). A specific class of diffeomorphisms, which define a subgroup structure in the underlying Lie group [28], can be parameterized by an ordinary differential equation (ODE) [6, 49]:

$$\frac{\mathrm{d}\Phi_t}{\mathrm{d}t} = V(\Phi_t), \tag{3}$$

where Φ_t is the diffeomorphism at time t, and V the stationary velocity vector field. *Forward warping:* by starting from the initial point (identity transformation) Φ_0, we are able to integrate (3) in time $(t : 0 \rightarrow 1)$ to obtain Φ_1, such that $F(\Phi_1(S)) \sim Z(S^Z)$. *Backward warping:* in general with learning warping transformations, integrating backward in time $(t : 1 \rightarrow 0)$ does not result in a reverse warping [6]. However, (3) defines a member of a Lie group, which provides a definition of the exponential operator. So, the correct way to define a backward warping Φ_{-1} is by integrating (3) over time $(t : 0 \rightarrow -1)$. To account for the richness of transformations, we parameterize the velocity V as a neural network, which gives a **Warping Neural ODE**, see Fig. 3.

3.2.2 Mechanisms for generating a GRF

Given the warping $\Phi(S)$, we need to make sure that $F(\Phi(S))$ is an isotropic GRF. While various divergences can be used, e.g., Jensen-Shannon [15] or KL [25], we simply minimize the Wasserstein (EM) distance [50] between distribution of warped images $F(\Phi(S))$ and GRF:

$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \inf_{\gamma \in \Pi\left(\mathbb{P}_r, \mathbb{P}_g\right)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|], \qquad (4)$$

where $\Pi\left(\mathbb{P}_r, \mathbb{P}_g\right)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. To achieve this, we minimize an efficient approximation of the Wasserstein distance similar to [5, 21]. However, in contrast to GAN, in our setup the generator (Neural ODE component) does not generate images based on random samples, but is only used to create a warping Φ with no randomness.

3.3. Summary of the procedure (with final loss)

Algorithm 1 Learning diffeomorphism $\Phi : F \to Z$

Input: General RF $F = \{F_i\}_{i=1}^{N_F}$ and GRF $Z = \{Z_i\}_{i=1}^{N_Z}$
Output: Diffeomorphism $\Phi(S)$

Require: parameterized by Neural Networks: V in (3), critic D (to minimize Wasserstein Distance), n_D number of critic's updates

while V has not converged **do**
  Set Φ_0 as identical transformation (vector field).
  Using Neural ODE(V, Φ_0) find a solution Φ_1.
  Given Φ_1, warp F to $\hat{F}$
  Run **MinWasDist($\hat{F}$, Z)** to minimize the Wasserstein distance
end while
procedure MinWasDist($\hat{F}$, Z)
  for $i = 0, \ldots, n_D$ **do**
    update D by minimizing *Critic's loss*:
      $-D(Z) + D(\hat{F}) + \lambda GP(D)$
      ▷ where GP(D) is gradient penalty for Critic D [21]
  end for
  update V by minimizing *ODE loss*:
    $-D(\hat{F})$ + JD + OG ▷ JD, OG defined below
end procedure

Remark 2. *While theoretically, it is guaranteed that there is a unique solution to the system (3) given Φ_0, see [43] (pp. 8), to accelerate convergence, we add constraints (penalties) to the ODE loss in Alg. 1, JD and OG respectively. Namely, we require (a) the Jacobian Determinant of each Φ_t to be non-negative [34], to avoid collapsing several pixels into one, and (b) prevent generating warping Φ_t, with vectors going outside the grid (image frame).*

$$JD = \sum_t \sum_s \Big(|JD(\Phi_t(s))| - JD(\Phi_t(s)) \Big)$$

$$OG = \sum_t \sum_s \Big((\text{grid}(s) + \Phi_t(s) - F_{\text{size}}) + (\text{grid}(s) + \Phi_t(s)) \Big)$$

Note that computation of OG term is motivated by our implementation of the warping Φ_t as a vector field, common in vision [6, 29], and considering the 'grid' as a mesh coordinate system from 0 to size of image F_{size}. Then, pixels in the 'grid(s)' will be sampled from (or move to) location $\text{grid}(s) + \Phi_t(s)$. The OG term prevents learning vector fields Φ_t, which map to outside of the grid. Alg. 2 describes the second part of the framework to select significant pixels on the source domain, given a learned warping Φ_t.

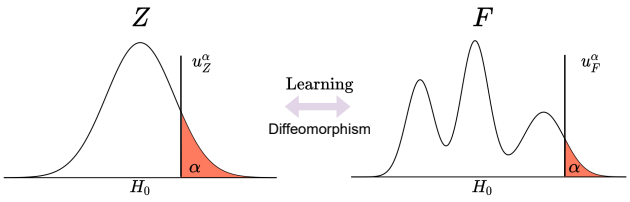

Figure 4. *Top:* Distribution of theoretical statistics $\mathbf{P}_{Z_{\max}}$ and $\mathbf{P}_{F_{\max}}$ under corresponding null hypotheses H_0 and thresholds u_Z and u_F, such that $\mathbb{P}\left(Z_{\max} \geq u_Z | H_0\right) = \alpha$ and $\mathbb{P}\left(F_{\max} \geq u_F | H_0\right) = \alpha$.

Algorithm 2 Selecting significant region $\mathcal{M}_F$

Input: RF $F = \{F_i\}_{i=1}^{N_F}$, learned diffeomorphism Φ_t
Output: Significant region $\mathcal{M}_F$

1: Apply forw. warping Φ_1 to F to generate $\hat{F}$ from the GRF.
2: Select significant pixels $\mathcal{M}_{\hat{F}}$ of $\hat{F}$ according to Thm. 3.1.
3: Apply rev. warping Φ_{-1} to $\mathcal{M}_{\hat{F}}$ to generate $\mathcal{M}_F$ on domain of F.

3.4. Applications

So far, we have discussed how to get a significant region for a general RF F, which corresponds to the rejection region of GRF, see Fig. 4. Our discussion was for a general case, without specifying how to obtain the RF F. Depending on how the RF F is obtained, this idea can be used for the two scenarios in §1: **(1)** to understand for which parts of the generated image, a model is the most uncertain and **(2)** to compare uncertainty between two models with different architectures. While (1) is important in scientific/healthcare settings, where we want to check whether we can trust a model in the region of interest, (2) helps evaluate whether users need to invest more in deploying a new model to decrease the uncertainty of their predictions.

Uncertainty within a model. To understand which part of the output image is the most uncertain, we generate F, given N outputs of the model, and using the variance per pixel. Thus, we have F, with $F(s)$ showing uncertainty per pixel. Since our goal is to decide which pixels are the most uncertain, we apply the Alg. (1) on the F and Z, generated under H_A. That is, all generated RF Z_i have some uncertain pixels, and on Fig. 4 we map only α regions between F and Z. Then, we find $\mathcal{M}_F$ as in Alg. 2.

Deriving 'significant regions' of uncertainty may appear similar to strong/weak class activations map methods [60]. However, our method is complimentary – it can be used downstream if the heat maps *also* include pixel-wise *confidence intervals* and satisfy GRRF assumptions.

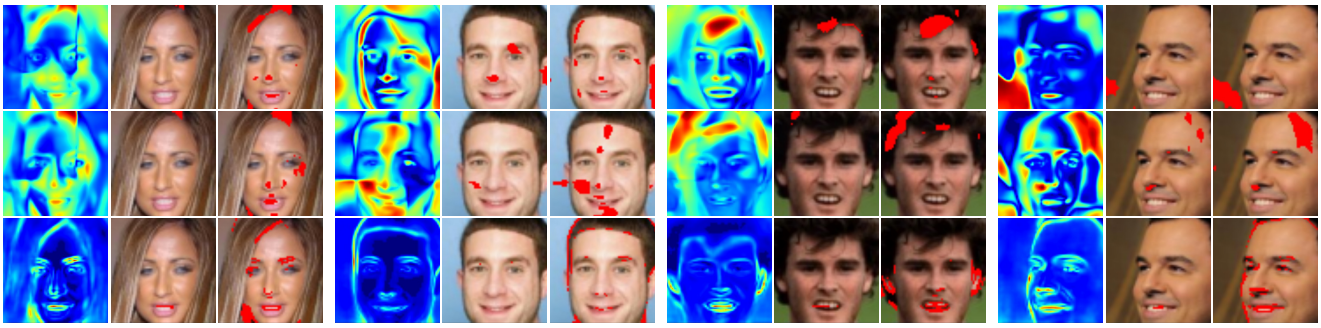

Figure 5. Rows: 1st – ResNet-18, 2nd – ResNet-34, 3d – ResNet50. Columns (by 3): 1) uncertainty map, generated by the VAE model with the ResNet base corresponding to the row, 2) significant uncertainty, derived by **our** method, 3) top 5% of uncertainty, typically used as significant uncertainty in vision.

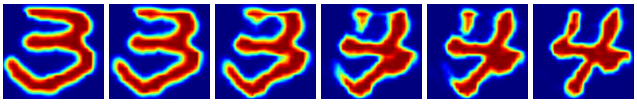

Figure 6. Continuous warping of '3' to '4' using our Warping NODE

Uncertainty between the models. Given two sets of images, we compute test statistics F^{obs} as mentioned before. Then, we use a bootstrap technique to construct a set of statistics $\{F_i\}_{i=1}^N$ (see supplement). We follow Alg. 1 to map the distribution F to Z completely, i.e., not just the α like in Scenario 1. Finally, we obtain significant region $\mathcal{M}_{F^{\text{obs}}}$ on F^{obs}. If $\mathcal{M}_{F^{\text{obs}}}$ contains at least a 1, then we reject null hypothesis that there are no improvements in uncertainty of the model in favor of H_A. While we provide a way to test the uncertainty *between* models, here, we will restrict our presentation to the uncertainty within a model.

Limitations: In the current form, our method cannot be used directly to prioritize deep uncertainty quantification approaches, e.g., BNNs [42], deep ensembles [35], and so on. Similar to hypothesis tests, we do not obtain a model ranking. But if we know, say the model/software cost, then the cheaper model is better *if* the H_0 is *not* rejected.

4. Experiments

We seek to demonstrate the ability of our model to provide estimates of statistical significance for the uncertainty generated by different probabilistic models, e.g., Variational Autoencoders [31], Neural Networks with MC Dropout [18], and Bayesian Neural Networks [22, 30, 42]. In our experiments, we use a broad range of common vision datasets, e.g. CelebA [37], AFHQ [11], KITTY [19], MS-COCO [36], and MR image data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) (three time-points, representing disease progression).

We construct significant regions according to $\mathbb{P}(F_{\max} \geq u_F | H_0) = 0.05$, a commonly used threshold [51]. For the baseline, we consider a typically used 5% quantile [30]. The appendix provides description of the DNN architectures as well as a simple experiment for different RFs with ground truth for significant regions.

Proof of concept: We start the experimental section by introducing Warping Neural ODE as a generative model, we train our model to warp samples from a distribution in the shape of handwritten digits '3' to digits '4'. Since we can evaluate the solution of ODE in (3) at arbitrary time t, in Fig. 6, we visualize the evolution of Φ_t. It is evident that our model can indeed learn a smooth diffeomorphism.

VAE: Given the generation mechanism of VAE [48, 59], the estimation of the epistemic uncertainty, i.e., uncertainty of the model, is direct. For each input image x, we run the inference M times generating M samples $x_1, \ldots, x_M$, on which we compute pixel-wise and channel-wise variance. Since the latent space of a VAE follows a Gaussian distribution, the resultant uncertainty F satisfies our assumptions (regarding GRRF), and so, we can apply our method directly to understand the significant regions.

For these experiments we consider different variations of VAE models (based on ResNet-18, ResNet-34 and ResNet-50 [24]) and four different datasets: CelebA [37], AFHQ [11]: closeups of 3 types of animals: Cat, Dog and Wild.

(a) *CelebA.* In Fig. 5, we show the uncertainty map and compare significant pixels derived from our method with the usual 5% quantile. *First*, we see that our approach picks up regions of clustered uncertainty, which indicates that our model is aware of spatial correlation in uncertainties.

Figure 7. Rows: cats, dogs, wild. Columns: significant uncertainty, derived by **our** method for different samples.

Figure 8. For each of the tree types of uncertainties along the rows (aleastoric, epistemic, and predictive) we demonstrate (left) the uncertainty of the depth estimation, (center) significant uncertainty region derived by our method, and (right) typically used as significant 5% quantile.

But, a standard 5% quantile picks up boundary points and stray/disperse points. It is especially obvious on the third row, for the most expressive network ResNet-50, where the generated uncertainty of the entire region is small. Our method picks up only the maximum and sensible regions (like teeth), while the 5% quantile picks up the boundary of the generated object – not a very meaningful region for calibration. *Second*, using the more expressive model (from top to bottom), our method picks up less significant uncertainty regions. In contrast, the 5% quantile picks up about the same number of pixels regardless of model confidence.

Observations: While we expect to have smaller significantly uncertain regions with an increase in complexity of the model, we do not expect models to be the most uncertain in the same exact regions. Moreover, we expect that with an increase in the complexity of the model, regions of significant uncertainty will be removed first. This can be seen by comparing uncertainty maps and significant pixels of our method in Fig. 5 across models. This behavior is harder to observe using the 5% quantile. We find that the process of elimination we observe is similar to 'backward' feature selection method in statistics [12], when we remove significant features based on p-values after each pass.

Computation/storage complexity: For CelebA, our model (14M parameters) occupies about 1934MiB. Runtime is 0.3s with a batch size of 1. On a standard system with four 2080TIs, 1 epoch of 10000 images needs 120s and full training (for warping) takes 7 hours. At test time, the hypothesis test (on PyTorch) is negligible ($\leq$ 1 ms).

(b) *AFHQ.* Since we showed the benefits of our method compared to the 5% quantile for selecting the significant uncertainty regions generated by VAE, for the AFHQ dataset,

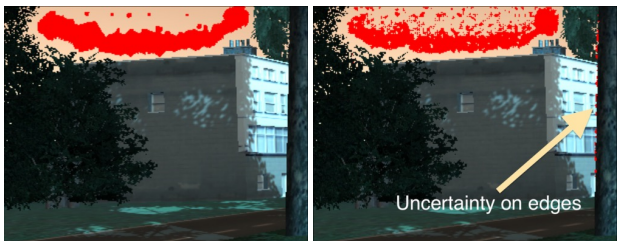

Figure 9. Zoomed region of predictive uncertainty, to show that in contrast to 5% qunatile (right), our method (left) does **not** pick up seldom (and meaningless) points on the edge of the tree.

we only provide results of our method for a single network ResNet-18 in Fig. 7. The goal is to show that significant uncertain regions are sensible. We see that the most uncertain regions are areas around eyes and ears.

MC dropout: In [18], the authors showed that when applying dropout on every layer, the dropout objective minimizes the Kullback–Leibler divergence between an approximate distribution and the posterior of a deep Gaussian process. Thus, uncertainty obtained from MC dropout satisfies our assumption of GRRF. Given a trained deterministic network, we can inject MC dropout layers to estimate uncertainty. We evaluate our method on uncertainty derived from MC dropout applied to two large scale datasets on different tasks: depth estimation and segmentation.

(a) *Depth estimation on Virtual KITTI dataset:* The virtual KITTI dataset [16] is a photo-realistic synthetic video dataset, which consists of high resolution scenes and is usually used for vision tasks such as object detection, multi-object tracking, scene-level and instance-level semantic segmentation, and depth estimation. We evaluate the ability of our model to pickup significantly uncertain pixels in high-resolution uncertainty maps. We follow the experiment setup of [22] to evaluate the depth of objects in images. Using MC-dropout, we generate uncertainty maps of size 320×1216 [18] and evaluate our model on three different types of uncertainties: epistemic, aleastoric and predictive (sum of both: epistemic and aleastoric). The results are shown in Fig. 8. Clearly, for all types of uncertainty (rows), our method (middle column) picks up regions of clustered uncertainty making the significance mask more smooth, indicating that our model is aware of spatial correlation in images, compared to the usual 5% quantile (right column). This is quite noticeable for epistemic uncertainty – uncertainty of the model (middle row). Further, Fig. 9 shows the predictive uncertainty of zoomed-in regions, for objects with strong edges, like a light pole or a tree. Our method does not pick up the edges as significant, making significant regions more meaningful and avoiding noise.

(b) *Segmentation on MS-COCO:* Common Objects in Context (COCO) [36] is a large-scale vision dataset that provides a rich set of visual descriptors and is widely used for baseline evaluations of semantic segmentation al-

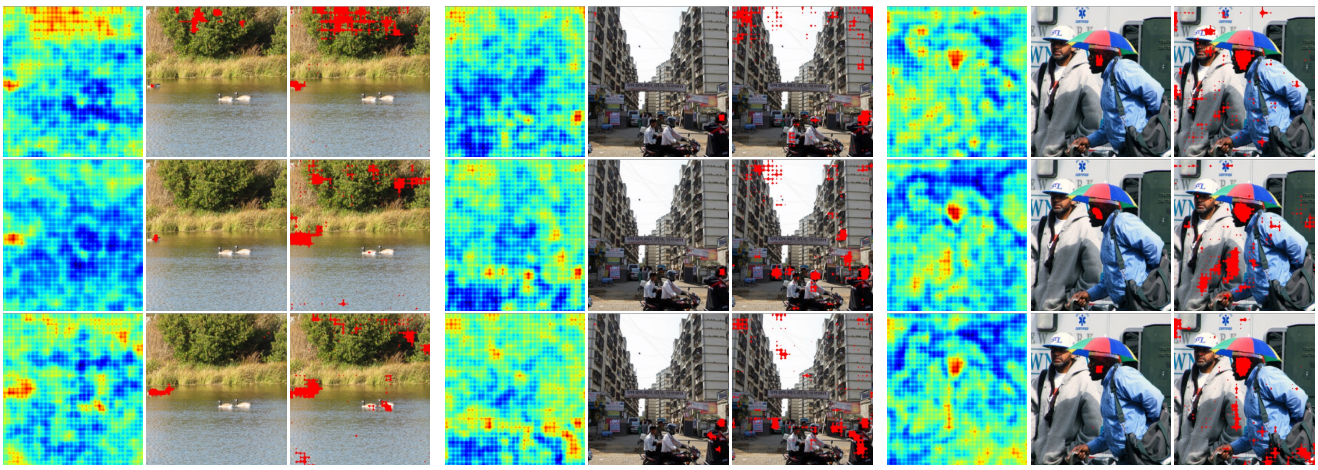

Figure 10. For each value of dropout probability p along the rows (0.01, 0.03, 0.04) we demonstrate by triplets: (left) the uncertainty of the pxiel-wise segmentation, (center) significant uncertainty region derived by our method, and (right) typically used as significant 5% quantile.

gorithms [9, 23]. To evaluate the effectiveness of our method, we generate segmentation uncertainty by applying MC Dropout to each layer of the DeepLab V3 [10] with the pre-trained checkpoint in PyTorch [44]. We measure uncertainties by summing over the individual variances of the model's predictions in softmax-normalized scores for each segmentation class and pixel-wise. To evaluate the effect of probability of dropout p on the generated uncertainty, we consider 3 variations: $p = 0.01$, $p = 0.03$ and $p = 0.04$. We noticed that using dropout with probability $p \geq 0.05$ leads to a very high uncertainty and no meaningful segmentations. Based on results in Fig. 10 for all 3 values of dropout probability p, there are 2 interesting observations. **(1)** As noted previously, the 5% quantile shows that significant uncertainty on one image is located in a lot of different classes and difficult to interpret. In contrast, our method is consistent when providing significant uncertainty within the class. **(2)** While the 5% quantile significant region changes through different values of p (across the rows in Fig. 10), the significant region based on our method is consistent across p. It means that ranging p does not change the significant uncertain region, which is reassuring.

Bayesian Neural Networks: The final method we use is Bayesian Neural Networks (BNN) [17, 39, 42]. We apply Temporal BNN to longitudinal (3 time point) brain imaging data obtained from Alzheimer's Disease Neuroimaging Initiative (ADNI), see supplement. The goal is to predict the brain image at the third time point, given that the first two steps are observed. To generate the uncertainty map, we collect predictions from 100 feed-forward runs of a trained BNN and compute voxel-wise standard deviations. Then we take a 2d slice as a final uncertainty map. We note that the application of a 5% quantile as a threshold for significance does *not* yield very meaningful results, completely covering two big regions of the brain: corpus callosum and caudate nucleus, and some stray pixels all around. In contrast, our method highlights small clustered regions. Since the data contains two clinically disparate groups, we should expect that samples from different groups have different significantly uncertain regions, generated by the predictive model. The 5% quantile threshold shows the same significant pixels, independent of diseased or control subjects (group difference testing). In contrast, our method nicely differentiates between CON and AD groups. In summary, we find that identifying statistically significant pixels shows that longitudinal progression in AD is quite different from CON, captured using our method (see supplement).

5. Conclusions

This paper provides a strategy for using existing deep neural network tools in conjunction with known results in Random Field Theory (RFT) to perform hypothesis tests on uncertainty maps from DNN models. Such a capability allows moving from subjective interpretation of uncertainties or the evaluation of deciles/quantiles to answering precisely stated hypotheses in a rigorous way. We believe that this capability is essential but currently missing and can further enable the use of DNN models from vision in mission-critical applications and for informing business/policy decisions.

Societal impacts. We provide a meaningful step towards interpreting/understanding uncertainty results from deep models in vision, a positive development from the standpoint of trustworthy AI models.

Acknowledgments

This work was supported by NIH grants RF1AG059312, RF1AG062336 and RF1AG059869, NSF award CCF 1918211 as well as funds from the American Family Insurance Data Science Institute at UW-Madison.

References

[1] Robert J Adler, Kevin Bartz, Sam C Kou, and Anthea Monod. Estimating thresholding levels for random fields via euler characteristics. *arXiv preprint arXiv:1704.08562*, 2017. 2, 3, 4

[2] Robert J Adler, Jonathan E Taylor, Keith J Worsley, and Keith Worsley. Applications of random fields and geometry: Foundations and case studies. In *In preparation, available on R. Adler's home*. Citeseer, 2007. 2, 3

[3] Zeynettin Akkus, Alfiia Galimzianova, Assaf Hoogi, Daniel L Rubin, and Bradley J Erickson. Deep learning for brain mri segmentation: state of the art and future directions. *Journal of digital imaging*, 30(4):449–459, 2017. 1

[4] Yasin Almalioglu, Muhamad Risqi U Saputra, Pedro PB de Gusmao, Andrew Markham, and Niki Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *2019 International conference on robotics and automation (ICRA)*, pages 5474–5480. IEEE, 2019. 1

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 5

[6] John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007. 4, 5

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015. 1

[8] J Cao and KJ Worsley. Applications of random fields in human brain mapping. In *Spatial Statistics: Methodological Aspects and Applications*, pages 169–182. Springer, 2001. 4

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 8

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 8

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 6

[12] Shelley Derksen and Harvey J Keselman. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2):265–282, 1992. 7

[13] Elizabeth A Eisenhauer, Patrick Therasse, Jan Bogaerts, Lawrence H Schwartz, Danielle Sargent, Robert Ford, Janet Dancey, S Arbuck, Steve Gwyther, Margaret Mooney, et al. New response evaluation criteria in solid tumours: revised recist guideline (version 1.1). *European journal of cancer*, 45(2):228–247, 2009. 2

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 1

[15] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE, 2004. 5

[16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 7

[17] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 8

[18] 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. PMLR, 2016. 6, 7

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 6

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 1

[21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 5

[22] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319, 2020. 6, 7

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 8

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[25] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007. 5

[26] HET Holgersson and Ghazi Shukur. Testing for multivariate heteroscedasticity. *Journal of Statistical Computation and Simulation*, 74(12):879–896, 2004. 1

[27] Christian Hubschneider, Robin Hutmacher, and J Marius Zöllner. Calibrating uncertainty models for steering angle estimation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1511–1518. IEEE, 2019. 2

[28] Arieh Iserles, Hans Z Munthe-Kaas, Syvert P Nørsett, and Antonella Zanna. Lie-group methods. *Acta numerica*, 9:215–365, 2000. 4

[29] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015. 5

[30] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017. 2, 6

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[32] Kishore Reddy Konda and Roland Memisevic. Learning visual odometry with a convolutional network. In *VISAPP (1)*, pages 486–490, 2015. 1

[33] Igor Kononenko. Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370, 1989. 1

[34] Dongyang Kuang. Cycle-consistent training for reducing negative jacobian determinant in deep registration networks. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 120–129. Springer, 2019. 5

[35] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017. 6

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6, 7

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 6

[38] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2014. 1

[39] David JC MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 354(1):73–80, 1995. 8

[40] J Huston McCulloch. Miscellanea on heteros* edasticity. *Econometrica (pre-1986)*, 53(2):483, 1985. 1

[41] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021. 1

[42] Jurijs Nazarovs, Ronak R Mehta, Vishnu Suresh Lokhande, and Vikas Singh. Graph reparameterizations for enabling 1000+ monte carlo iterations in bayesian deep neural networks. In *Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence*, volume 2021. NIH Public Access, 2021. 1, 6, 8

[43] Carl Öhrnell. Lie groups and pde, 2020. 5

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 8

[45] Ryan Poplin, Avinash V Varadarajan, Katy Blumer, Yun Liu, Michael V McConnell, Greg S Corrado, Lily Peng, and Dale R Webster. Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature Biomedical Engineering*, 2(3):158–164, 2018. 1

[46] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *arXiv preprint arXiv:1609.08976*, 2016. 1

[47] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[48] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pages 14866–14876, 2019. 6

[49] François Rousseau, Lucas Drumetz, and Ronan Fablet. Residual networks as flows of diffeomorphisms. *Journal of Mathematical Imaging and Vision*, 62(3):365–375, 2020. 4

[50] Ludger Rüschendorf. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129, 1985. 5

[51] John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003. 6

[52] David L Streiner and Geoffrey R Norman. Correction for multiple testing: is there a resolution? *Chest*, 140(1):16–18, 2011. 2

[53] Jonathan Taylor, Akimichi Takemura, Robert J Adler, et al. Validity of the expected euler characteristic heuristic. *Annals of Probability*, 33(4):1362–1396, 2005. 3

[54] Jonathan E Taylor et al. A gaussian kinematic formula. *Annals of probability*, 34(1):122–158, 2006. 3

[55] Jonathan E Taylor and Keith J Worsley. Detecting sparse signals in random fields, with an application to brain mapping. *Journal of the American Statistical Association*, 102(479):913–928, 2007. 4

[56] Tien Vo, Akshay Mishra, Vamsi Ithapu, Vikas Singh, and Michael A Newton. Dimension constraints improve hypothesis testing for large-scale, graph-associated, brain-image data. *Biostatistics*, 02 2021. kxab001. 2

[57] Keith J Worsley, Alan C Evans, Sean Marrett, and P Neelin. A three-dimensional statistical analysis for cbf activation studies in human brain. *Journal of Cerebral Blood Flow & Metabolism*, 12(6):900–918, 1992. 2, 3

[58] Martina Zähle. Lipschitz-killing curvatures of self-similar random fractals. *Transactions of the American Mathematical Society*, 363(5):2663–2684, 2011. 3

[59] Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. D-vae: A variational autoencoder for directed acyclic graphs. *arXiv preprint arXiv:1904.11088*, 2019. 6

[60] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 5