Supplementary Materials:
Multi-view 3D Object Reconstruction and Uncertainty Modelling with Neural Shape Prior

Ziwei Liao and Steven L. Waslander
Robotics Institute & Institute for Aerospace Study
University of Toronto
ziwei.liao@mail.utoronto.ca, steven.waslander@utoronto.ca

S1. Uncertainty in Downstream Tasks

We have carried out experiments to prove the effectiveness of uncertainty-aware shape models in reconstruction and multi-view fusion. Besides, our work opens up the possibilities for multiple downstream tasks which we defer to future work.

Uncertainty for perception. Since our uncertainty-aware shape model is continuous and differentiable, we can propagate the uncertainty stored in the shape into any observation model with a given math formulation. For example, differentiable rendering [6] is the core technology to constrain neural implicit representation [7, 10], and solve perception tasks such as camera pose estimation [15], object shape and pose estimation [1], and object-level SLAM [14, 13]. They are the core abilities for robotics, VR & AR, and autonomous driving. In Figure S1, we show a simple example of propagating the uncertainty in the 3D shape model into any given camera view with differentiable rendering. We sample multiple latent codes and calculate the sample mean and variance of the rendered depth of each latent code. Our rendered uncertainty map can be used in the differentiable constraints, which naturally assigns weights for areas with high information, to fuse multi-view observations in challenging real scenarios with occlusion and ambiguity.

Uncertainty for planning and control. For indoor robots or autonomous driving cars operating in real scenarios, their world model (e.g., a map of objects) is uncertain with partial, noisy, and limited observations. Our model explicitly quantifies the uncertainty and stores it in the shape model, which can be used by the downstream modules to make decisions. In motion planning [9], uncertainty helps safely navigate and stay away from uncertain objects to avoid obstacle. In robotic grasping [4], uncertainty can efficiently guide the manipulator to find a next-best-view to better reconstruct the uncertain part of the shape to guarantee the success rate of the grasps. Our model has the potential to offer an uncertainty source for those tasks.

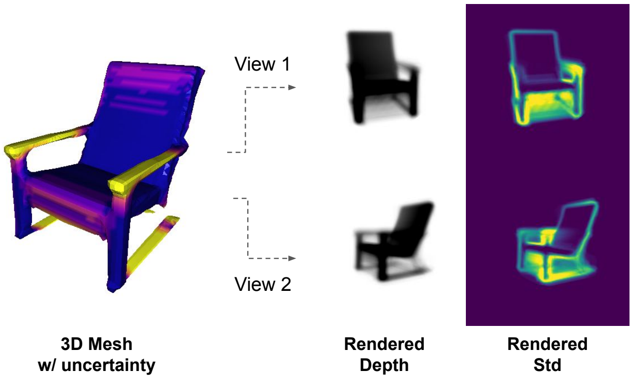

Figure S1. Differentiable rendering with uncertainty. We can propagate the uncertainty in 3D shapes into any camera view with differentiable rendering. The rendered uncertainty (std) can be used by downstream tasks, such as constructing losses and propagating gradients for pose and shape optimization and object-level SLAM.

S2. Uncertainty Analysis

S2.1. Calibration Plot

We have evaluated the effectiveness of using uncertainty on reconstruction and multi-view fusion. We further evaluate the estimated uncertainty quality of correlating with the accuracy. An estimated uncertainty is *well-calibrated* when the estimation errors are lying into the predicted uncertain threshold. We introduce how we draw the calibration plot in Sec S2.2.

We present the calibration plot on the Pix3D dataset in Figure S2, where the X axis shows the predicted probability, and the Y axis counts the real frequency where the estimation errors lying within the predicted uncertain threshold. A well-calibrated plot should be close to the line $Y = X$, where the real frequency aligns perfectly with the predicted probability. Our method outputs uncertainty in latent space and the signed distance space, as in Figure 2. The uncertainty in the latent space is directly output from the Encoder,

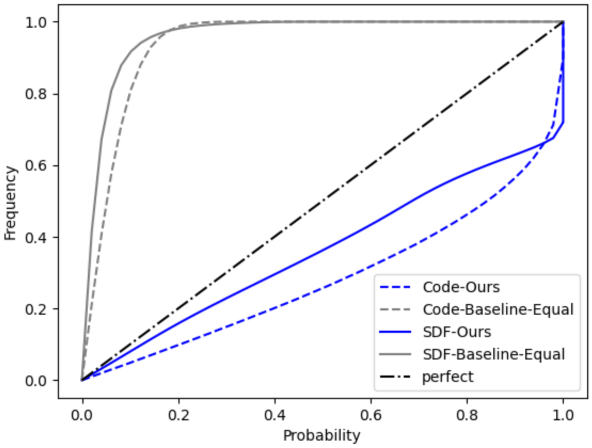

Figure S2. The calibration plot on the Pix3D dataset. The plots for the uncertainty of the latent space and SDF space are shown separately. We further show a baseline outputting equal uncertain of 1. Our approach is siginficantly better calibrated (closer to $Y = X$ line) than the baseline.

and then propagates using Monte-carlo sampling through the decoder to the signed distance space, as described in Sec 3.2. We draw calibration plots for both of them. We use a baseline with an equal variance of 1 for each code dimension. We can see that it is non-trivial to output a well-calibrated uncertainty for both the latent code and the 3D SDF values. The calibration plots of the baseline method are far away from the line of $Y = X$, while ours is significantly closer, which shows a much better calibrated result than the baseline. At the same time, we still see an opportunity for further improvement, e.g., with temperature scaling [3]. We hope our approach can serve as a benchmark for future research that can output better uncertainty for 3D neural shape models.

S2.2. Drawing a Calibration Plot

We present a calibration plot in Sec S2.1. We describe in detail here the method to draw a calibration plot for the latent space and the signed distance function space. Given N 1-dimensional random variables $\{X_i\}_{i=1}^N$, for each X_i, our algorithm outputs the estimated univariate Gaussian distributions with the mean and variance of $\{(\mu_i, \sigma_i^2)\}_{i=1}^N$. To draw a calibration plot, we first uniformly sample T probability p_t in the range of $(0, 1)$. For each p_t, we count the real frequency F_t that the estimation errors $\{|X_i - \mu_i|\}_{i=1}^N$ lying inside the predicted threshold r_i^t:

$$F_t = \frac{1}{N} \Sigma_{i=1}^N I_{|X_i - \mu_i| \leq r_i^t} \tag{1}$$

where I_C is an indicator function outputting 1 when C holds, and 0 otherwise. The predicted threshold r_i^t can be calculated using the quantile function, which is the inverse cumulative distribution function. For a normal distribution, the cumulative distribution function calculates the probability p of a random variable X lies inside a given range x:

$$F_X(x) := Pr(|X| \leq x) = p \tag{2}$$

The quantile function inversely calculates the range of x that the variable X lies inside with a given probability p:

$$Q(p) = F_X^{-1}(p) = x \tag{3}$$

For a normal distribution, the value of $Q(p)$ can be directly queried given any p. For example, $Q(0.9984) \approx 3$ for a normal Gaussian, which is the 3-σ principle. For a Gaussian distribution with mean and variance of (μ_i, σ_i^2), we can calculate the predicted threshold $r_i^t = Q(p_t)\sigma_i$ by multiplying the standard deviation. Then we can count the real frequency F_t according to Eq. 1. Finally, we can draw a plot with the sampled probability $\{p_t\}_{t=1}^T$ as the X axis, and the counted frequency $\{F_t\}_{t=1}^T$ as the Y axis. When drawing the calibration plot, we consider each dimensions of each latent codes independently, and also consider each SDF values independently.

S2.3. Metric Evaluation

We quantitatively use two metrics, Energy Score (ES) and Negative Log Likelihood (NLL) to evaluate the calibration and sharpness of the estimated SDF uncertainty in Table S1. We show results for the points near the object's surface (Surface), far away from the surface (Non-Surface) by a distance threshold of 0.01, and all points (All Space) separately. We ablade two models trained with ES and NLL separately. Considering the lack of an uncertainty-aware shape model in the literature, we use a baseline that outputs equal uncertainty of 1. As expected, our two uncertainty models both outperform the baseline by an order of magnitude, which shows that the estimated uncertainty contains much more valid information. Compared with the NLL model, the ES model can output slightly better uncertainty, but an obviously better accuracy of the 3D reconstruction. We notice the uncertainty of the points near the surface (Surface) is better calibrated than those far away from the surface (Non-Surface). The points near the surface are critical for the process of Marching Cubes to reconstruct objects' mesh and they model the uncertainty of the mesh surface. We think this is because, during the training of the Decoder, more points are sampled near the surface where the Decoder is more capable of estimating the SDFs, which benefits the uncertainty estimation too.

S2.4. Distribution of the SDF Values

We propagate a Gaussian distribution in the latent space through the Decoder to generate the distributions of SDF values. However, the Decoder, with multiple non-linear activation layers, is a highly non-linear function. Thus, the

	All Space		Surface		Non-Surface		Reconstruction
Methods	ES ↓	NLL ↓	ES ↓	NLL ↓	ES ↓	NLL ↓	IoU ↑
Equal Var	0.1932	-0.203	0.1914	-0.210	0.1939	-0.201	/
Ours-NLL	0.0353	-2.720	0.0235	-3.065	0.0399	-2.587	0.299
Ours-ES	**0.0338**	**-2.753**	**0.0221**	**-3.095**	**0.0384**	**-2.621**	**0.335**

Table S1. Uncertainty analysis in the 3D SDF space. Ours trained with ES shows better performance compared with Ours trained with NLL model, and much better compared with the baseline outputting equal uncertainty of 1.

SDF distributions are not guaranteed to be Gaussian again. We show the distribution of the SDF values of 4 points with different distances to the object surface in Figure S3. Their distributions are calculated through Monte-carlo sampling with a sample num of 1000. We can see that for the points far away from the surface (the two figures in the bottom), the distributions of the SDFs value only have one mode and can approximately form a Gaussian distribution. We notice an interesting findings for the SDF values near the surface (the two figures in the top). They do not follow a Gaussian distribution perfectly but have a peak much larger than other areas in the negative areas near the surface. We assume this is a bias introduced during training where the Decoder is trained with unbalanced samples of points inside and outside surface. The origin DeepSDF paper [10] truncates SDF values inside the surface to -0.01. The trained Decoder will tend to output a narrow range of negative values, while a much larger range output space for positive values.

There are several ways to model the distribution of the SDF values depending on the downstream tasks. e.g., in the qualitative evaluation, we use the sample variance of each vertices of the mesh to approximately colorize it with uncertainty. For other tasks that need to accurately propagate uncertainty, e.g., differentiable rendering, a parametric approximation of the SDF distibution, e.g., Gaussian, will make the process easier to tackle. It will be interesting future work to explore more expressive distribution for SDF values, e.g., Gaussian-mixture models [11].

S3. More Qualitative Results

We show more results of multi-view fusion in Figure S4. When fusing views with more information about the objects, our method decreases the uncertainty of the reconstructed shape and gets better results than the baseline. We notice interesting findings that some semantic parts of the objects, e.g., arms, legs, backs, and thin structures, can be well identified by uncertainty. We also notice that for the first example, half of the 10th view including the chair arm is occluded and is very challenging, so fusing this view slightly increases the uncertainty of the arm. It will be valuable future work to deeply investigate the relationship among the semantic parts of 3D shapes, the occluded image areas, and the uncertainty in the encoded shape latent space.

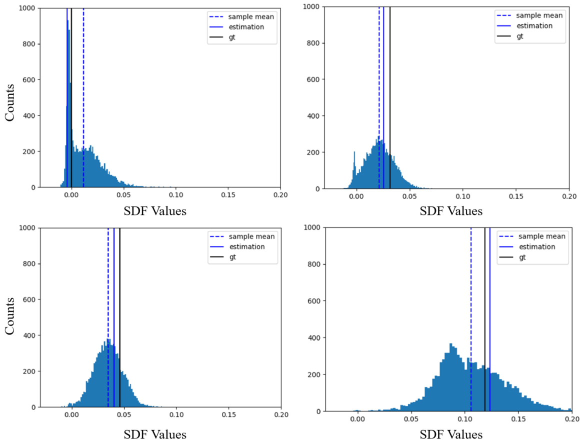

Figure S3. Distribution of the SDF values after propagating the latent code distribution to the SDF space with Monte-carlo sampling. Four points with distance near and far from the surface (SDF=0) are shown. The groundtruth value (gt), estimated value (estimation) and the sample mean are shown. The estimation is close to the groudntruth value and the distribution covers the error.

S4. Computation Analysis

We show the inference time with a resolution of 128 for the decoder and a number of $N = 10$ for the Monte Carlo sampling on an A100 GPU in Table S2. To generate the uncertainty for a mesh, only a small computation overhead is needed for the mesh vertices (2K points). However, if necessary, the uncertainty for a whole SDF field is more expensive, which consists of 128^3 (2M) points. We inherit the design of DeepSDF [10] as Decoder, which is the bottleneck of our computation. Depending on the downstream tasks, we can use multiple parameters as trade-off between effectiveness and efficiency, such as the sampling times and shape resolution. Although the Monte-Carlo sampling approach used to propagate uncertainty through the decoder to the SDF grid is computationally expensive, we prove it is possible to estimate uncertainty with the decoder fixed, and leave it as future work further improvements in efficiency for global SDF uncertainty estimation., Methods such as local linearization of the nonlinear decoder, or making the decoder to direct output uncertainty can be used in this context to reduce this computational cost. For multi-view fusion, thanks to the fusion in the latent space, we only need 0.1

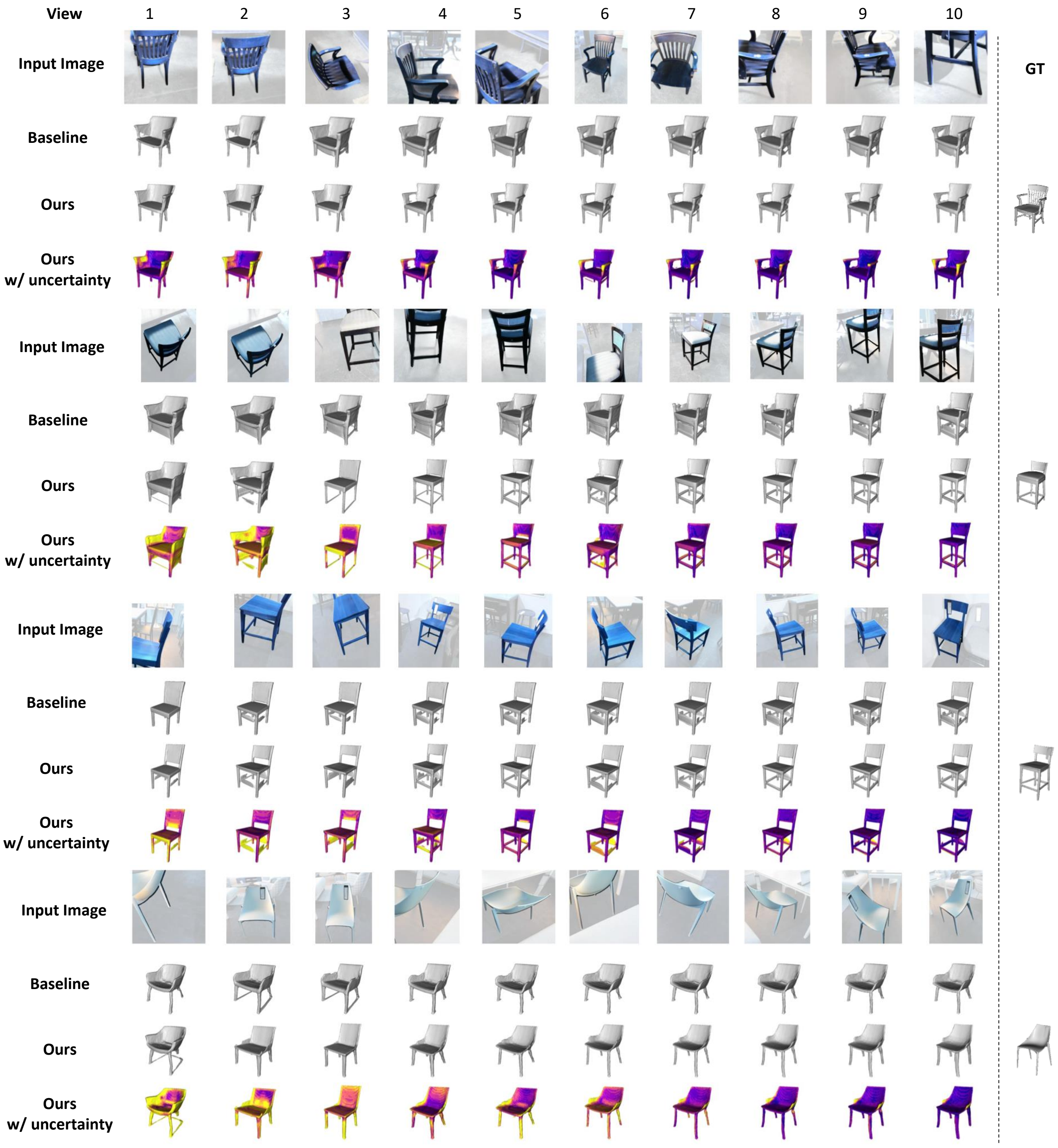

Figure S4. Qualitative results of multi-view fusion on the Pix3D dataset (part 2). Results after fusing 1 to 10 input images are given.

ms to fuse 10 views and get the final fused latent code with uncertainty.

S5. Discussion and Future Work

Multi-categories. In this work, we train each category (chairs, and tables) independently and separately, as the same as the previous work (DeepSDF [10], FroDO [12]). With the success of large language models [8, 5] to show the effectiveness of the scaling up, and the availability of new 3D object datasets [2] much larger than ShapeNet, it will be interesting to explore an uncertainty-aware general object model that can be used for multiple categories of ob-

Modules	SDF Field	Mesh	Mesh w/ Uncer.	SDF Field w/ Uncer.
Encoder	0.06			
Decoder	0.49			
Marching Cubes	-	0.13	0.13	-
Decode Uncer.	-	-	0.001 x 10	-
Decode Uncer.	-	-	-	0.49 x 10
Total	0.55	0.68	0.69	5.45

Table S2. Computation analysis (Unit: s). We sample $N = 10$ times during Monte-carlo sampling. When generating mesh with uncertainty, only the mesh vertices (around 2K) are propagated. When generating SDF field with uncertainty, 128^3 (2M) points are propagated.

jects.

Decoupling the shape decoder and the image encoder. We take the training strategy of first training the decoder, and then fixing the decoder to train the encoder, instead of end-to-end training. In this way, the decoder acts as an independent shape representation in 3D that has no relationship with input modalities. Thus, we can easily support new conditional modalities by training a new encoder, and leverage the prior knowledge stored in the latent space of the shape decoder. It remains valuable future work to extend to other modalities, e.g., texts, depth maps, pointclouds, by training a new corresponding encoder and fusing the results in the latent space in a similar way to the multi-view fusion.

References

[1] Leonard Bruns and Patric Jensfelt. Sdfest: Categorical pose and shape estimation of objects from rgb-d using signed distance fields. *IEEE Robotics and Automation Letters*, 7(4):9597–9604, 2022.

[2] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[3] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[4] Liren Jin, Xieyuanli Chen, Julius Rückin, and Marija Popović. Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. *arXiv preprint arXiv:2303.01284*, 2023.

[5] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[6] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020.

[7] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[8] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[9] Èric Pairet, Juan David Hernández, Marc Carreras, Yvan Petillot, and Morteza Lahijanian. Online mapping and motion planning under uncertainty for safe navigation in unknown environments. *IEEE Transactions on Automation Science and Engineering*, 19(4):3356–3378, 2021.

[10] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[11] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[12] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14720–14729, 2020.

[13] Edgar Sucar, Kentaro Wada, and Andrew Davison. Nodeslam: Neural object descriptors for multi-view shape reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 949–958. IEEE, 2020.

[14] Jingwen Wang, Martin Rünz, and Lourdes Agapito. Dsp-slam: Object oriented slam with deep shape priors. In *2021 International Conference on 3D Vision (3DV)*, pages 1362–1371. IEEE, 2021.

[15] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020.