Uncertainty Estimation for 3D Dense Prediction via Cross-Point Embeddings

Kaiwen Cai[1], Chris Xiaoxuan Lu[2], and Xiaowei Huang[1]

***Abstract*—Dense prediction tasks are common for 3D point clouds, but the uncertainties inherent in massive points and their embeddings have long been ignored. In this work, we present CUE, a novel uncertainty estimation method for dense prediction tasks in 3D point clouds. Inspired by metric learning, the key idea of CUE is to explore cross-point embeddings upon a conventional 3D dense prediction pipeline. Specifically, CUE involves building a probabilistic embedding model and then enforcing metric alignments of massive points in the embedding space. We also propose CUE+, which enhances CUE by explicitly modeling cross-point dependencies in the covariance matrix. We demonstrate that both CUE and CUE+ are generic and effective for uncertainty estimation in 3D point clouds with two different tasks: (1) in 3D geometric feature learning we for the first time obtain well-calibrated uncertainty, and (2) in semantic segmentation we reduce uncertainty's Expected Calibration Error of the state-of-the-arts by 16.5%. All uncertainties are estimated without compromising predictive performance.**

***Index Terms*—Probabilistic Inference, Computer Vision for Automation, Semantic Scene Understanding**

I. Introduction

The process of predicting a label of each point in a point cloud is known as 3D dense prediction. It is a crucial aspect of robotic perception and autonomy, enabling tasks such as semantic segmentation, depth completion, and scene flow estimation.

UNet-based networks have become the de-facto choice for point cloud dense prediction [1]–[4]. In a UNet-like network, the input and the output of two correspondingly linked layers have the same number of points, e.g., if the input point cloud is represented by a $N \times 3$ tensor, then the output of its correspondingly linked layer is a $N \times D$ tensor. In this regard, the output can also be considered as an embedding map, and a dense prediction network can then be decomposed as an embedding learning network and a task-specific regressor (or classifier). Thus, the core of the dense prediction task is embedding learning.

Embedding learning aims to learn a discriminative model that maps samples of the same class closer together and those of different classes farther apart in the embedding space. Successful embedding learning facilitates many downstream tasks, including image retrieval [5], face recognition [6] and zero-shot learning [7]. In addition to enhancing the discriminative capability of the embedding model, quantifying its uncertainty is also gaining significant attention.

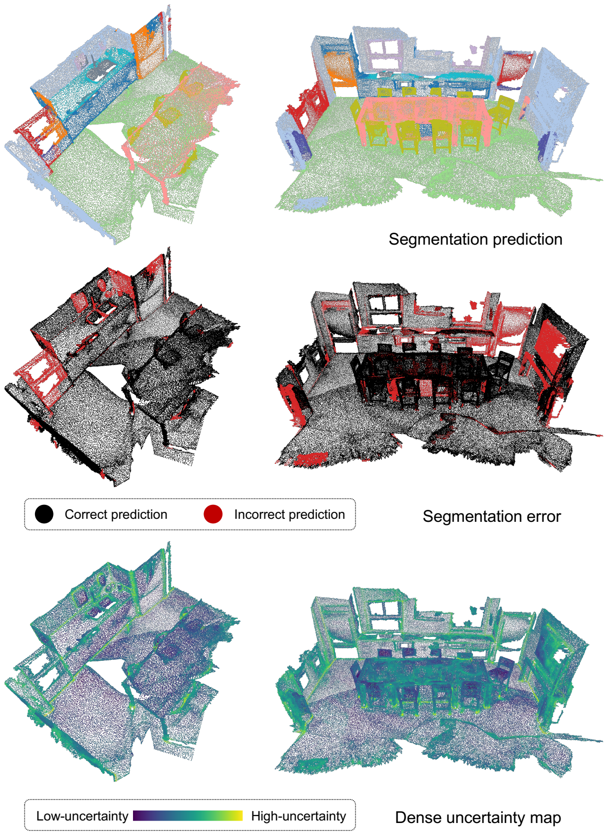


Fig. 1. In a 3D dense prediction task, i.e., 3D semantic segmentation, we present the segmentation prediction (top), segmentation error (middle) and dense uncertainty map (bottom, estimated by CUE+) of two scenes from ScanNet validation split. Incorrect predictions tend to have high uncertainties.

For dense prediction tasks of point clouds, it is beneficial that an uncertainty level could be provided in conjunction with the point-wise labels to make its downstream decision-making more information-aware. For example, in an autonomous vehicle scenario where semantic labels of each point on the road are predicted, an estimated uncertainty level would aid the vehicle in determining when to trust the prediction and optimize planning and control. Such promising benefits have stimulated the development of various uncertainty estimation methods for different dense prediction tasks, including (1) using the output of the logit layer to calculate softmax entropy [8], (2) building a two-head network to predict the mean and variance of an embedding separately [9], and (3) resorting to a Bayesian Neural Network (BNN) model with Monte Carlo Dropout (MCD) to approximate posterior weights [10].

However, two major issues persist in current methods for estimating uncertainty in the dense prediction of 3D point clouds. Firstly, points can only interact within the limited receptive field of convolution kernels and require a shared MLP to facilitate im-

Manuscript received: Sep. 22, 2022; Revised Jan. 24, 2023; Accepted Feb. 20, 2023. This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

[1] Kaiwen Cai and Xiaowei Huang are with Department of Computer Science, University of Liverpool, England {k.cai, xiaowei.huang}@liverpool.ac.uk

[2] Chris Xiaoxuan Lu is with School of Informatics, University of Edinburgh, Scotland xiaoxuan.lu@ed.ac.uk

Digital Object Identifier (DOI): see top of this page.

plicit interactions among logits. This under-treatment of cross-point dependencies, unfortunately, often results in suboptimal uncertainty estimation as evidenced by [11]. Secondly, a notable trait of the prevalent dense prediction networks is that they are sequential compositions of embedding learning networks and task-specific regressors (or classifiers). While prior arts have shown that incorporating embedding learning in regression or classification tasks can yield better predictive performance [12], [13], it is largely under-explored if utilizing embedding learning can also give rise to better-calibrated uncertainty.

In this paper, we propose a novel and generic uncertainty estimation pipeline, called CUE in the paper for **C**ross-point embedding **U**ncertainty **E**stimation, to bridge the gap between the dense prediction of point clouds and its uncertainty quantification. CUE involves constructing a probabilistic embedding model and enforcing metric alignments of massive points in the embedding space. To address the aforementioned issues, CUE emphasizes the importance of embedding learning, and exploits this embedding space through a diagonal multivariate Gaussian model that facilitates cross-point interactions. Additionally, we propose CUE+ that further utilizes cross-point dependencies by a low-rank multivariate Gaussian model. The low-rank covariance matrix in CUE+ explicitly captures off-diagonal elements' dependencies while maintaining computational efficiency. Our specific contributions are:

- We propose a generic framework for estimating uncertainty in dense prediction tasks of 3D point clouds.
- We propose a novel approach that fully leverages cross-point information for estimating uncertainty.
- We validate our proposed method on two representative dense prediction tasks, with the experimental results consistently demonstrating that our method produces better-calibrated uncertainty than state-of-the-art methods without compromising predictive performance.
- Source code of both CUE and CUE+ is available at: https://github.com/ramdrop/cue.

II. Related Work

A. Dense Prediction of 3D Point Cloud

Given the dense nature of the 3D point cloud, we focus on dense prediction tasks, e.g., 3D geometric feature learning and 3D semantic segmentation.

3D Geometric Features Learning: To find the correspondences between point clouds in the absence of relative transformation information, a number of methods are to convert point clouds from the 3D Euclidean space to a feature space, where correspondences are identified as the nearest neighbors. Early work rely on hand-crafted features [14] to perform this conversion. Recently, deep learned geometric features are becoming increasingly popular, which are typically based on volumetric and pointwise operations on point clouds: (1) Volumetric methods such as 3DMatch [15] and FCGF [2] learn patch and point descriptors respectively by applying a 3D Convolutional Neural Network (CNN) on volumetric input. (2) Point-wise methods such as PointNet [16] use parallel shared MLP to learn global or dense features, and DGCNN [17] combines pointwise MLP with dynamic graph neural networks to obtain flexible and effective feature extractors for unordered point clouds. Nevertheless, these methods primarily focus on improving predictive performance while ignoring the uncertainty inherent in massive points.

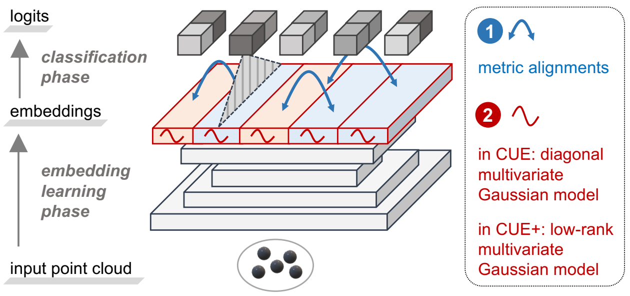


Fig. 2. An overview of the proposed CUE and CUE+. We take semantic segmentation for instance where there are 5 points in the input point cloud and 2 classes in the labels. CUE explores cross-point embeddings by building a probabilistic embedding model (Red curves) and enforcing metric alignments (Blue arrows), and CUE+ goes further by replacing the diagonal covariance matrix with a low-rank covariance matrix.

3D Semantic Segmentation: PointNet [16] is the very first work for 3D point cloud learning, while its shared-MLP architecture shows strong representation capability, it ignores the local context. Following works propose different solutions to make for this limitation: PointNet++ [18] adopts hierarchical sampling strategies, KPConv [4] proposes a kernel-based MLP operation mimicking convolution, MinkowskiNet [19] extends 3D convolution to 4D space and develops sparse operation library for point clouds, and PointTransformer [20] shows the power of Transformer mechanisms in point cloud processing.

B. Uncertainty Estimation

Embedding Learning Uncertainty: Kendall [9] categorizes uncertainties in deep learning as two types: aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty arises from data noises, while epistemic uncertainty refers to model uncertainty which can be reduced with sufficient training data. Embedding learning is commonly applied to image recognition tasks, where most methods focus on estimating aleatoric uncertainty. For example, PFE [21] models face embeddings as Gaussian distributions and uses the proposed Mutual Likelihood Score to measure the likelihood of two embeddings belonging to the same class. DUL [22] proposes to learn aleatoric uncertainty for both regression and classification face recognition tasks. BTL [23] proposes a Bayesian loss to learn aleatoric uncertainty in place recognition. RUL [24] uses relative uncertainty measurements to learn aleatoric uncertainty.

The image recognition tasks discussed above involves learning a single feature for an entire image, however, in the dense prediction task of point clouds, thousands of features (i.e., equals to the number of points in the point cloud) need to be learned for a single point cloud. Furthermore, image recognition is applied to regular-size images, while point clouds are totally unordered and have a varied size. The large number of features within a batch and irregular input size render it rather challenging to estimate uncertainty for a 3D point cloud.

Semantic Segmentation Uncertainty: Popular uncertainty estimation methods for semantic segmentation, as outlined in [25], include softmax entropy [8] , BNN [26], learned aleatoric uncertainty [9] , auxiliary network [27] and variance propagation based on Assumed Density Function (ADF) [28]. While these approaches are widely used, they often assume independence between pixels or points. This lack of consideration for cross-pixel or cross-point dependencies can lead to less accurate uncertainty estimation [11].

Embedding learning has been explored in in the context of image segmentation with studies such as [13] and [29] demonstrating that contrastive learning can optimize embedding space and improve prediction performance in semantic segmentation tasks. Research like [12] also supports the idea that optimized embeddings can contribute to improved predictive performance. However, these studies primarily focus on using embedding learning to enhance prediction performance, rather than for estimating uncertainty. SSN [11] utilizes a low-rank multivariate Gaussian model to account for cross-pixel dependencies, but it is developed for logits and does not incorporate embedding optimization.

The proposed CUE is based on a probabilistic embedding model and enforces metric alignments in the embedding space by using bayesian triplet loss. Bayesian triplet loss has been used in [23] for image recognition, but with some key differences. Firstly, the image recognition [23] requires a single embedding for an entire image (i.e., a set of all pixels), while massive point-wise embeddings are desired in CUE. This means that while traditional 2D CNNs can be used to extract image features, an efficient and effective network for extracting both point features and uncertainties remains unknown. Thus, we investigate efficient networks and propose sampling strategies for utilizing cross-point embeddings of massive and unordered points in a batch; Secondly, the probabilistic embedding model of [23] ignores the cross-point dependencies, whereas the proposed CUE+ addresses this issue by using a low-rank multivariate Gaussian model.

III. Method

A. Probabilistic Embedding Model

A dense prediction network maps a batch of points to a set of scalars. This process can be broken down into a metric learning phase and a task-oriented regression or classification phase. Specifically, given a point cloud $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{N\times 3}$, where N is the number of points, the network f_θ maps it to a set of embeddings $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N\times D}$, where D is the embedding dimension. The metric learning phase can be written as $\boldsymbol{\mathcal{X}} = f_\theta(\boldsymbol{\mathcal{P}})$. Then, a task-oriented regressor (or classifier) f_r generates predictions $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N\times 1}$ (or $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N\times C}$ where C is the number of class) for the set of embedding $\boldsymbol{\mathcal{X}}$. This regression or classification phase can be written as $\boldsymbol{\mathcal{Y}} = f_r(\boldsymbol{\mathcal{X}})$. In the above formulation, predictions are treated as deterministic and do not take into account the inherent noise from the data. In comparision, a probabilistic prediction model (e.g., probabilistic semantic segmentation [9]) represents the predictions as a Gaussian distribution, which provides uncertainty level in addition to the prediction. However, the embeddings are still treated deterministic and are given equal weight, meaning each embedding contributes equally to the regressor (or classifier).

Inspired by the use of probabilistic contrastive learning in face recognition [21], [22], we build a probabilistic embedding model for a point cloud, with embeddings represented by a diagonal multivariate Gaussian distribution, which can be written as

$$\boldsymbol{\mathcal{X}} \sim \boldsymbol{\mathcal{N}}\left(\boldsymbol{\mu}, \boldsymbol{\Lambda}\right), \tag{1}$$

where $\boldsymbol{\mu} = f_\mu(\boldsymbol{\mathcal{P}}) \in \mathbb{R}^{N\times D}$ and $\boldsymbol{\Lambda} = f_\sigma(\boldsymbol{\mathcal{P}}) \in \mathbb{R}^{(N\times D)\times(N\times D)}$ is a diagonal matrix. f_μ and f_σ representss the mean branch and variance branch of the network f_θ, respectively. We will later propose a full-covariance multivariate Gaussian model and demonstrate its superiority in Sec. III-B2.

B. Metric Alignments of Embeddings

Once the probabilistic embedding model has been constructed, the next step is to optimize the embedding space and obtain the uncertainty.

A traditional probabilistic prediction pipeline [9] only allows for implicit interactions between logits through a shared multi-layer perceptron (MLP), i.e., there is no explicit interaction within layers. In contrast, the above probabilistic embedding model allows for enhanced interactions of logits and the estimatimation of uncertainties. An overview of the proposed CUE and CUE+ is presented in Fig. 2, where CUE explores cross-point embeddings by constructing a probabilistic embedding model and enforcing metric alignments, and CUE+ goes further by using a diagonal covariance matrix in place of a low-rank covariance matrix. In the following discussion, we will first describe CUE which is based on the diagonal multivariate Gaussian model and then present an improved version, CUE+, which is based on the low-rank multivariate Gaussian model.

1) CUE: Given a triplet of samples, $\{\boldsymbol{P_a}, \boldsymbol{P_p}, \boldsymbol{P_n} | \boldsymbol{P_i} \in \mathbb{R}^{1\times 3}, i = a, p, n\}$, their embeddings are obtained as $\{\boldsymbol{X_a}, \boldsymbol{X_p}, \boldsymbol{X_n} | \boldsymbol{X_i} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \boldsymbol{\mu}_i \in \mathbb{R}^{1\times D}, \boldsymbol{\Sigma}_i \in \mathbb{R}^{1\times D}, i = a, p, n\}$, where the subscripts a, p, n denote an anchor, positive and negative sample, respectively. In the probabilistic setting, we are interested in the probability of the positive embedding being closer to the anchor than the negative one:

$$P(\|\boldsymbol{X}_a - \boldsymbol{X}_p\| - \|\boldsymbol{X}_a - \boldsymbol{X}_n\| + m < 0). \tag{2}$$

Rewrite it as

$$P(\tau < -m), \tag{3}$$

where the new distribution $\tau = \sum_{d=1}^{D} \boldsymbol{T}^d = \sum_{d=1}^{D}(\boldsymbol{X}_a^d - \boldsymbol{X}_p^d)^2 - (\boldsymbol{X}_a^d - \boldsymbol{X}_n^d)^2$, and d denotes d^{th} dimension . According to central limit theorem, τ will approximate a normal distribution when D is large, i.e.,$\frac{\tau-\mu_\tau}{\sigma_\tau} \sim \mathcal{N}(0,1)$, where μ_τ and σ_τ^2 are the mean and the variance of the distribution τ. Then (3) is solved as $P(\tau < -m) = \Phi_{\mathcal{N}(0,1)}(\frac{-m-\mu_\tau}{\sigma_\tau})$, where Φ is the Conditional Density Function (CDF). The task now is to find an analytical solution of μ_τ and σ_τ. The mean $\mathbb{E}'[\tau]$ and variance $\mathbb{D}'[\tau]$ of a single dimension is given as (the superscript d at the right-hand side is omitted for brevity)

$$\begin{aligned}\mathbb{E}[\boldsymbol{T}^d] &= \mu_p^2 + \sigma_p^2 - \mu_n^2 - \sigma_n^2 - 2\mu_a\left(\mu_p - \mu_n\right),\\ \mathbb{D}[\boldsymbol{T}^d] &= 2[\sigma_p^4 + 2\mu_p^2\sigma_p^2 + 2\left(\sigma_a^2 + \mu_a^2\right)\left(\sigma_p^2 + \mu_p^2\right) - 2\mu_a^2\mu_p^2\\ &\quad - 4\mu_a\mu_p\sigma_p^2] + 2[\sigma_n^4 + 2\mu_n^2\sigma_n^2 + 2\left(\sigma_a^2 + \mu_a^2\right)\left(\sigma_n^2 + \mu_n^2\right)\\ &\quad - 2\mu_a^2\mu_n^2 - 4\mu_a\mu_n\sigma_n^2] - 8\mu_p\mu_n\sigma_a^2.\end{aligned} \tag{4}$$

Since the embedding model is assumed to be isotropic, we arrive at

$$\mu_\tau = \sum_{d}^{D} \mathbb{E}[\boldsymbol{T}^d], \quad \sigma_\tau^2 = \sum_{d}^{D} \mathbb{D}[\boldsymbol{T}^d]. \tag{5}$$

In summary, after the network generates a set of embeddings for a point cloud, we calculate the probability of the positive

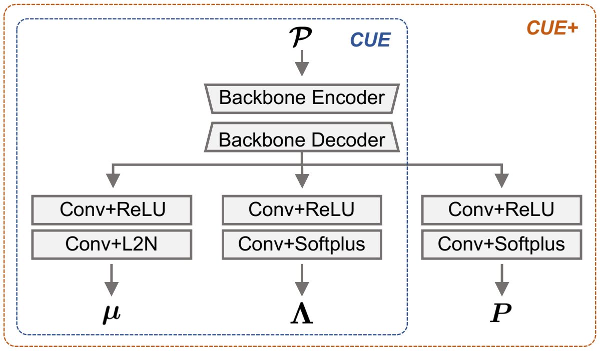


Fig. 3. The network architectures of CUE and CUE+: $\boldsymbol{\mathcal{P}}$ represents a 3D point cloud, $\boldsymbol{\mu}$ the embeddings' mean, $\boldsymbol{\Lambda}$ the diagonal elements of embeddings' covariance matrix, $\boldsymbol{P}$ the scale factor of embeddings' covariance matrix.

embedding being closer to the anchor than the negative one, and the goal is to minimize the metric loss derived from (2):

$$L_M = -\frac{1}{T}\sum_{t=1}^{T}\log P(\|\boldsymbol{X}_{t,a}-\boldsymbol{X}_{t,p}\|-\|\boldsymbol{X}_{t,a}-\boldsymbol{X}_{t,n}\|<-m) \tag{6}$$

where T is the number of total triplets in a mini-batch.

2) CUE+: Points usually have a spatial correlation with their neighbors. For example, points at the boundaries of an object usually exhibit high uncertainty since the points around the boundary have varied semantic labels. However, CUE fails to model point-wise dependencies because the diagonal covariance matrix of CUE (see (1)) is based on the assumption that points are independent of each other. To address this issue, we propose a solution to further capture the point-wise dependencies by a full-covariance multivariate Gaussian model. Specifically, the diagonal covariance matrix in (1) is replaced with a full covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{(N\times D)\times(N\times D)}$

$$\boldsymbol{\mathcal{X}} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{7}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{N\times D}$. However, the computational complexity of the full covariance matrix $\boldsymbol{\Sigma}$ scales with the square of N, and a point cloud usually consists of tens of thousands of points, i.e., $N > 10^4$. This makes training networks difficult. To alleviate this issue, we resort to a low-rank parameterization of the covariance matrix [30]

$$\boldsymbol{\Sigma} = \boldsymbol{P}\boldsymbol{P}^T + \boldsymbol{\Lambda}, \tag{8}$$

where the scale factor $\boldsymbol{P} \in \mathbb{R}^{(N\times D)\times K}$ and K is the rank of the parameterization, $\boldsymbol{\Lambda} \in \mathbb{R}^{(N\times D)\times(N\times D)}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix. The pipeline based on a low-rank covariance matrix is named CUE+ as it goes beyond CUE by learning parameters of additional elements other than diagonal elements of the covariance matrix. This allows for explicit description of point-wise dependencies through the learned variances, making it a more powerful model than CUE.

For ease of application, we follow [30] and choose $K = 1$. L_M is then used to train the network. By experimental results, we show that CUE+ generates better-calibrated uncertainty than CUE (see Sec. IV).

The network architectures of the proposed CUE and CUE+ are shown in Fig. 3, where $\boldsymbol{\mathcal{P}}$ represents a 3D point cloud. The backbone encoder and decoder can be chosen from any UNet-like network. We add three branches to predict the mean $\boldsymbol{\mu}$, the diagonal covariance matrix $\boldsymbol{\Lambda}$ and the scale factor $\boldsymbol{P}$. As L2-Normalization has been shown to improve prediction performance [2], we add an L2-Normalization layer to the end of the $\boldsymbol{\mu}$ branch. For the variance branch, we follow the practice of [23] and add softplus layers at the end of the $\boldsymbol{\Lambda}$ and $\boldsymbol{P}$ branches.

C. Cross-point Embedding Sampling

In the previous section, we discussed two methods to enforce metric alignments on the triplets of point embeddings. In this section, we will explain how these triplets are constructed.

In 3D dense prediction tasks, we categorize cross-point embeddings into two types: 1) Cross-point Embeddings within a Single point cloud (CES) and 2) Cross-point Embeddings among Multiple point clouds (CEM). Depending on the specific 3D dense prediction tasks, we explore different types of embeddings as follows.

1) CES: [29] proposed to improve point cloud segmentation performance by applying contrastive learning to point boundaries. Inspired by this approach, we propose a method of exploring CES in 3D semantic segmentation tasks by focusing on objects' boundaries: we first randomly sample anchors from the point cloud $\boldsymbol{\mathcal{P}}$, and then, within the neighbors of each anchor $\{\boldsymbol{X}_{a,i}|i = 1, 2, .., T\}$, choose embeddings with the same class label as positives $\{\boldsymbol{X}_{p,i}|i = 1, 2, .., T\}$, and those with different class labels as negatives $\{\boldsymbol{X}_{n,i}|i = 1, 2, .., T\}$. The triplets are constructed as $\{\{\boldsymbol{X}_{a,i}, \boldsymbol{X}_{p,i}, \boldsymbol{X}_{n,i}|i = 1, 2, .., T\}\}$.

2) CEM: [2] presents embedding learning with triplet loss, and we demonstrate exploring CEM in 3D geometric feature learning tasks by following its sampling method. Specifically, given point clouds $\boldsymbol{\mathcal{P}}_i$ and $\boldsymbol{\mathcal{P}}_j$ and the relative transformation $\boldsymbol{\mathcal{T}}$, we first construct a list of all positive pairs from which we randomly draw T pairs $\{(\boldsymbol{X}_{i_t,a}, \boldsymbol{X}_{j_t,a}), t = 1, 2, .., T\}$. Then we randomly choose the negative samples of each point as $\{(\boldsymbol{X}_{i_t,a}, \boldsymbol{X}_{i_t,n}), t = 1, 2, .., T\}$ and $\{(\boldsymbol{X}_{j_t,a}, \boldsymbol{X}_{j_t,n}), t = 1, 2, .., T\}$. Finally, The triplets are obtained as $\{(\boldsymbol{X}_{i_t,a}, \boldsymbol{X}_{j_t,a}, \boldsymbol{X}_{i_t,n}), t = 1, 2, .., T\}$ and $\{(\boldsymbol{X}_{j_t,a}, \boldsymbol{X}_{i_t,a}, \boldsymbol{X}_{j_t,n}), t = 1, 2, .., T\}$.

Sampling is a crucial component in making CUE and CUE+ possible for accurate uncertainty estimation in 3D dense prediction. Methods for uncertainty estimation in image recognition, such as [22]–[24], cannot be directly applied to 3D dense prediction as they are designed for global feature learning and not capable of handling large amounts of embeddings in 3D point clouds.

IV. Experimental Results

A. 3D Geometric Feature Learning

3D geometric feature learning aims to train a deep neural network to map raw points in Euclidean space to a feature space, with the goal of having points with similar geometric characteristics be close to each other in the feature space. [2] studied different sampling strategies, including hardest-triplet sampling and random triplet sampling, and used triplet loss as the loss function. We build upon its model by exploring CEM with the sampling method described in Sec.III-C2 and adapting their conventional triplet loss to our metric loss L_M.

Datasets. We use the 3D Match dataset [15], following the official training and evaluation splits.

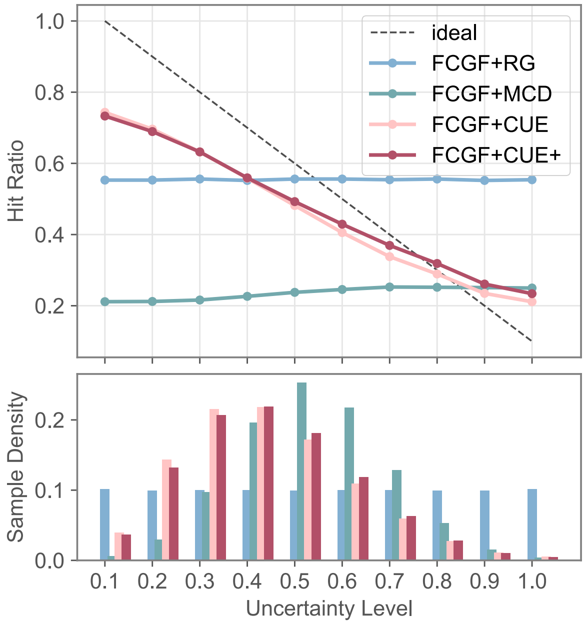


Fig. 4. Reliability diagram on the 3D Match Benchmark. CUE and CUE+ are closer to the ideally-calibrated line than others.

TABLE I
PREDICTIVE PERFORMANCE AND UNCERTAINTY QUALITY ON THE 3D MATCH BENCHMARK.

Method	FMR@0.05 ↑	ECE ↓
FPFH* [14]	36.4	\
PerfectMatch* [31]	94.9	\
FCGF* [2]	95.3	\
SpinNet [3]	97.5	\
FCGF [2]	97.5	\
FCGF+RG	97.5	0.251
FCGF+MCD	94.1	0.344
FCGF+CUE	97.5	0.142
FCGF+CUE+	97.6	0.135

* denotes predicting correspondences without a symmetric test [32].

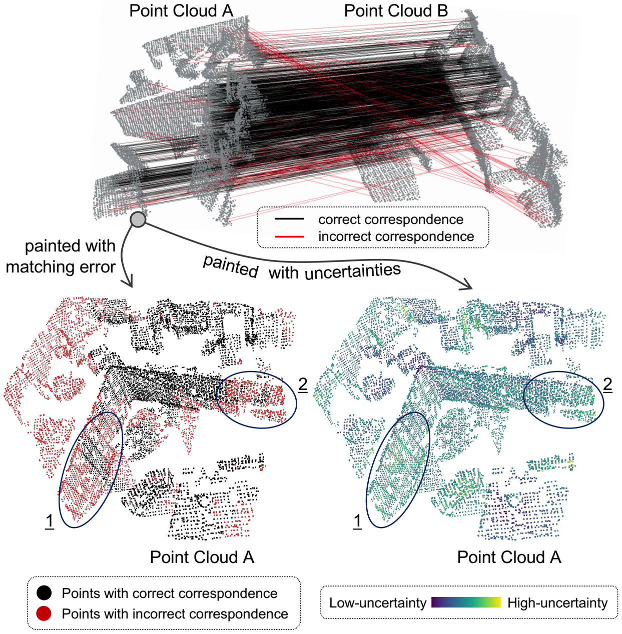
Model Architectures. FCGF [2] is a 3D convolutional network that is the first to integrate metric learning in a fully-convolutional setting. We choose FCGF [2] as our backbone because it has state-of-the-art predictive performance, fast training and inferencing. To add the ability to estimate the uncertainty of each point, the FCGF is integrated with the proposed CUE and CUE+, as is shown in Fig. 3.

Training Details. We train FCGF following the original paper [2], i.e., Hardest-contrastive loss, 100 epochs with SGD optimizer and batch size 4, learning rate starts from 0.1 with exponential decay rate 0.99, dada augmentation includes random scaling $\in [0.8, 1.2]$ and random rotation $\in [0°, 360°)$.

Competing Methods.

- Random Guess (RG): After training the FCGF, each point is assigned a random uncertainty value drawn from a uniform distribution.
- MCD: We insert dropout layers with dropout rate $p = 0.1$ after every convolutional layer. We take $N = 40$ samples from the weights' posterior distribution at test time.
- CUE: To maintain the original predictive performance, we freeze the $\boldsymbol{\mu}$ branch and train $\boldsymbol{\Lambda}$ branches with L_M.
- CUE+: We freeze the $\boldsymbol{\mu}$ branch, and train $\boldsymbol{\Lambda}$ and $\boldsymbol{P}$ branches with L_M.

Note that MCD produces epistemic uncertainty, while our methods generate aleatoric uncertainty. MCD is included in this comparison for the sake of completeness.



a) Matching Error b) Dense uncertainty map

Fig. 5. Matching results and dense uncertainty map (estimated by CUE+) of a point cloud from the 3D Match Benchmark. Incorrect correspondences (area 1 and 2) tend to have high uncertainties.

Evaluation Metrics. To evaluate the predictive performance, we use Feature Matching Recall with $0.1m$ inlier distance threshold and 0.05 inlier recall threshold (FMR@0.05) [2]. We adopt the widely used Expected Calibration Error (ECE) [23] and the reliability diagram [23] to evaluate uncertainty quality, where we calculate the Hit Ratio [2] of points in the same bin.

Results. We evaluate the above methods on the 3D Match Benchmark [15]. We establish correspondences by the nearest neighbor search in the embedding space, with each correspondence having an estimated uncertainty[1]. Table. I shows the predictive performance and uncertainty quality of different methods on the 3DMatch dataset. MCD shows degraded predictive performance due to the dropout layers negatively impacting the network's representation ability. Besides, the ECE of MCD is even worse than RG, meaning MCD fails to provide a sensible uncertainty. Since the $\boldsymbol{\mu}$ branch is inherited from the backbone network, CUE and CUE+ do not sacrifice predictive accuracy. Compared with RG, CUE reduces ECE by 43.4%. CUE+ shows similar predictive performance as CUE, but reduces the ECE of CUE by 4.9%.

Fig. 5 shows the matching results and dense uncertainty map estimated by CUE+ of a point cloud. We can observe that incorrect correspondences (area 1 and 2) tend to have high uncertainties. Fig. 4 presents the reliability diagram on the 3D Match Benchmark. The ideal line indicates that points with higher uncertainty levels should have lower hit ratios. It can be seen that the cuves of RG and MCD are nearly flat, indicating them are not able to effectively associate uncertainty levels with hit ratios. The CUE and CUE+ are shown to have much closer lines to the ideal in the reliability diagram. In low-uncertainty regions (Uncertainty Level $\leq$ 0.4), the performance of CUE

[1] We follow the covariance formulation in [23] and use the sum of two points' uncertainty as the correspondence's uncertainty.

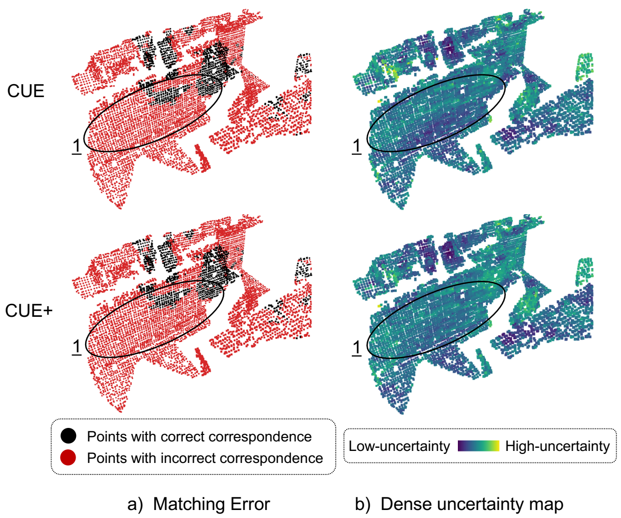


Fig. 6. Matching results and dense uncertainty map of a point cloud from the 3D Match Benchmark. The first row and second row denote results generated by CUE and CUE+, respectively. CUE is overconfident about the incorrect correspondences (area 1), while CUE+ is properly confident.

TABLE II
PREDICTIVE PERFORMANCE AND UNCERTAINTY QUALITY ON THE SCANNET VALIDATION SPLIT.

	Method	mIOU ↑	ECE ↓
Without uncertainty estimation	PointNet [16]	0.535	\
	PointConv [35]	0.610	\
	KPConv deform [4]	0.692	\
	SparseConvNet [36]	0.693	\
	Mink [19]	0.715	\
With uncertainty estimation	Mink+SE [25]	0.715	0.251
	Mink+AU [9]	0.717	0.254
	Mink+DUL [22]	0.719	0.173
	Mink+RUL [24]	0.712	0.187
	Mink+MCD(p=0.20)	0.658	0.176
	Mink+MCD(p=0.05)	0.663	0.170
	Mink+CUE	0.721	0.142
	Mink+CUE+	0.727	0.141

and CUE+ are similar, but in high-uncertainty regions (0.5 ≤Uncertainty Level ≤ 0.7), CUE is less effective than CUE+ due to overconfidence. The trend is shown in Fig. 6, where CUE is overconfident about the incorrect correspondences (1 areas), while CUE+ is properly confident. This demonstrates that the estimated uncertainty is highly practical, as it can be utilized as an effective tool for filtering incorrect correspondences when performing point cloud registration.

In summary, both the proposed CUE and CUE+ can estimated uncertainty for 3D geometric feature learning without compromising predictive performance, and CUE+ shows better calibrated uncertainty than CUE.

B. 3D Semantic Segmentation

3D semantic segmentation aims to learn a classification network that predicts the class of each point in a point cloud. We explore CES with sampling methods described in III-C1.

Datasets. Following [33], we use the ScanNet dataset [34] and evaluate models on the ScanNet validation split.

Model Architectures. We choose MinkowskiNet42 (Mink) [19], [33] as our 3D semantic segmentation backbone since it has high accuracy and low inference latency. The semantic segmentation network is the same as that in Fig. 3, except that we add a convolution layer as the segmentation classifier before the L2-Normalization layer of the $\boldsymbol{\mu}$ branch.

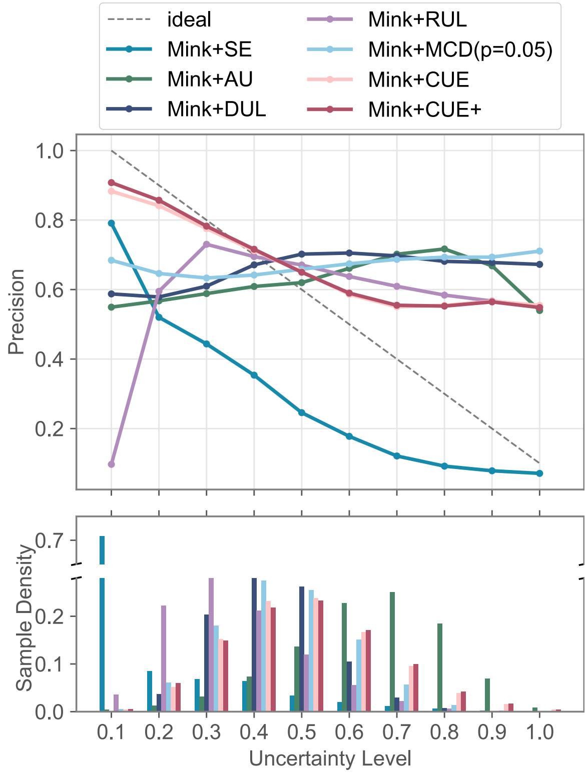


Fig. 7. Reliability diagram on the ScanNet validation split. CUE and CUE+ are closer to the ideally-calibrated line than other methods.

Training Details. We train the model for 10^5 steps with an SGD optimizer, learning rate starting from 0.1 with a cosine annealing schedule and a linear warmup. We use a batch size of 8. More training details can be found in reference [33].

Competing Methods. We evaluate the performance of CUE and CUE+ against several well-known uncertainty estimation methods for image recognition or segmentation:

- Softmax Entropy [25] (SE): The uncertainty is calculated based on the entropy of softmax output as $H = -\sum_c^C p_c \log(p_c)/log(C) \in [0, 1]$, where C is the number of classes, p_c is a probability by the softmax layer.
- Aleatoric Uncertainty [9], [25] (AU): The Logit are modeled as a Gaussian distribution, with the mean and the variance predicted by two seperate heads of the network. We use MC sampling (with $n = 10$) to generate samples from the logits distribution and optimize the network with the cross entropy loss.
- Data Uncertainty Learning (DUL) [22]: DUL is designed for face image recognition. We adapt it to our 3D semantic segmentation task by incorporating its distributional representation [22] and replacing L_M with its $\mathcal{L}_{cls}$.
- Relative Uncertainty Learning (RUL) [24]: RUL was proposed for facial expression recognition. We include its *feature mixture* of [24] and replace L_M with its L_{total}.
- MCD [25]: MCD estimates the epistemic uncertainty as enabling dropout at the test time approximates a random sampling of the model's weights. Test time inference is obtained by $p_c = \frac{1}{N}\sum_n^N p_{n,c}$, where p_c denotes the output of the Softmax layer. We set the number of MC samples $N = 40$ as suggested by [37]. We evaluate MCD with two dropout probability settings: $p = 0.2$ and $p = 0.05$. Since

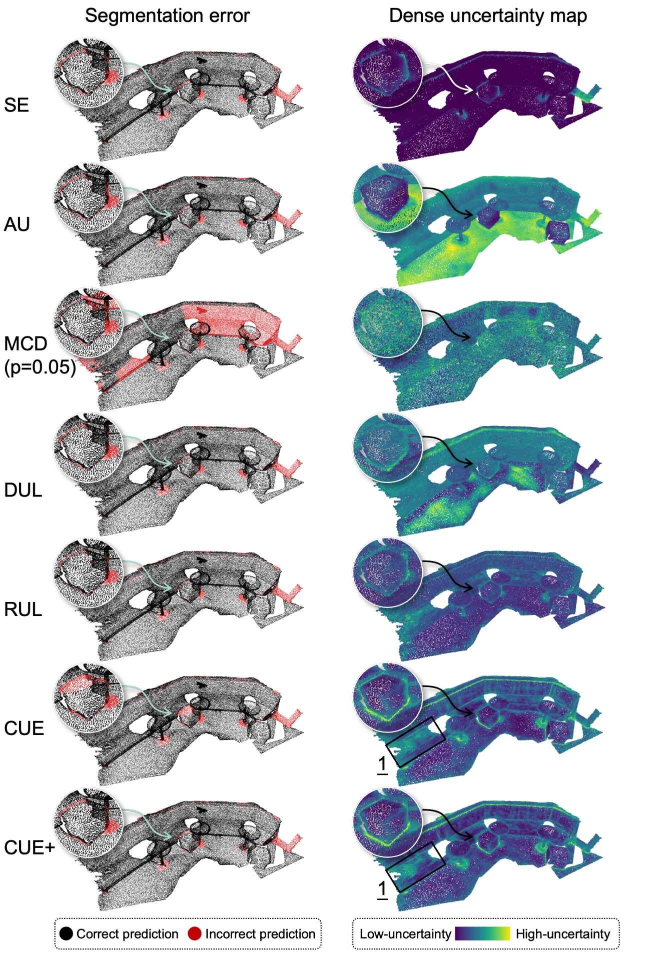


Fig. 8. Segmentation errors (left column) and dense uncertainty maps (right column) on a scene from the ScanNet validation split. CUE and CUE+ produce better-calibrated dense uncertainty maps than others. For correct predictions (rectangular area 1), CUE is under-confident while CUE+ is more confident than CUE.

AU and MCD generates high-dimensional variance vectors, the variance vectors are transformed into uncertainty levels by $y(1-0.5q)+(1-y)(0.5q)$, where $q \in [0,1]$ is the normalized variance [25].

- CUE / CUE+: We train the CUE / CUE+ network from scratch with a weighted sum of the cross entropy loss and the metric loss $L = L_{CE} + \lambda L_M$, where we set $\lambda = 1$ for all experiments.

Evaluation Metrics. Mean Intersection over Union (mIoU) is a commonly used metric to evaluate the performance of image segmentation models. It is calculated as the ratio of the intersection of the ground-truth labels and predicted labels to their union. A higher mIoU indicates better performance. Additionally, the reliability diagram [23] and ECE [23] are used to evaluate uncertainty quality, where we calculate the precision of the points in each bin.

Results. Table. II presents the predictive performance and uncertainty quality on the ScanNet validation split. As is shown in the 'Without uncertainty estimation' section, Mink has the highest mIOU, indicating that it is the state-of-the-art 3D segmentation model. In the 'With uncertainty estimation' section, we observe that SE, AU, DUL, RUL and CUE provide comparable predictive performance to Mink, while CUE+ promotes Mink with the most significant boost of 0.012 in mIoU. However, MCD shows degraded performance, which is attributed to the fact that dropout layers decrease the model's representative power. Regarding uncertainty quality, SE and AU show the highest ECE, which are 0.251 and 0.254, respectively. DUL, RUL and MCD indicate similar ECE results, which are 0.173, 0.187, and 0.176(p=0.20)/0.170(p=0.05). In comparison, CUE and CUE+ provide significantly improved uncertainty with the ECE 0.142 and 0.141. This means CUE+ reduces the ECE of the best existing method, MCD(p=0.05), by 16.5%. The results indicate that existing uncertainty estimation methods designed for image recognition (SE, AU, DUL and RUL) cannot produce satisfactory results on 3D dense prediction tasks as they fail to capture cross-point relations, particularly in a batch with massive points. Even though MCD shows relatively better uncertainty quality, this is achieved at the cost of predictive performance and processing time.

TABLE III
COMPUTATIONAL COMPLEXITY

Method	Num of Params ↓	Processing time/ Point cloud ↓
Mink [19]	36.88M	0.163s
Mink+SE [25]	36.88M	0.165s
Mink+AU [9]	36.90M	0.172s
Mink+DUL [22]	36.90M	0.172s
Mink+RUL [24]	36.90M	0.165s
Mink+MCD	36.88M	6.507s
Mink+CUE	36.90M	0.173s
Mink+CUE+	36.91M	0.174s

Fig. 7 shows the reliability diagram on the ScanNet validation split. It can be seen that only SE, CUE and CUE+ show descending trends as the ideal line, while the other methods present opposite or inconsistent trends. However, SE deviates significantly from the ideal line, while CUE is close to the ideal line and CUE+ improves CUE in the low-uncertainty region. Fig. 1 presents the qualitative results of CUE+, where we can observe a significant correlation between segmentation prediction error and estimated uncertainty, i.e., Incorrect predictions tend to have high uncertainties.

Fig. 8 presents segmentation errors and dense uncertainty maps by different methods on the ScanNet validation split. For incorrect predictions (black points in the magnified area), we can observe that SE fails to detect them and shows high confidence, while CUE and CUE+ are uncertain about those incorrect predictions. AU and DUL are under-confident in most areas, while RUL is over-confident in many points with incorrect predictions, e.g., the corners of the magnified areas and the table legs. And MCD (p=0.05) cannot produce sensible results. For correct predictions (Rectangular area 1), CUE is under-confident while CUE+ is more confident than CUE.

Table. III presents the computation complexity of the proposed CUE/CUE+ and comparing methods. We use NVIDIA A100 when evaluating networks' processing time during the inference phase. It is shown that Mink has the least trainable network parameters and the fastest processing speed. Although MCD does not add additional parameters to Mink, multiple forward propagations significantly increase MCD's processing time. AU, DUL, CUE have more parameters since they have more layers, which brings 6% overhead to Mink. Because CUE+ has an additional branch, it has slightly more parameters and processing time compared to CUE. Overall, CUE and CUE+

bring little overhead to the original network.

The above results indicate that CUE and CUE+ provide better-calibrated uncertainty than existing methods without compromising predictive performance, and CUE+ outperforms CUE in both predictive performance and uncertainty quality with marginally more overhead. This demonstrates that explicitly expressing cross-point embedding interactions contributes to uncertainty estimation in 3D dense prediction tasks, where a low-rank multivariate Gaussian model is more effective than a diagonal one.

V. Conclusion

Observing the fact that dense prediction networks are sequential compositions of embedding learning networks and task-specific regressors (or classifiers), we propose CUE that estimates uncertainty by building a probabilistic embedding model and enforcing metric alignments with a diagonal multivariate Gaussian model. We further propose CUE+ that enhances cross-point interactions with a low-rank multivariate Gaussian model, which explicitly expresses off-diagonal elements' dependencies while maintaining computational efficiency. Experimental results on the 3D Match Benchmark and the ScanNet dataset have shown that CUE and CUE+ are generic and efficient (bring negligible overhead) tools for uncertainty estimation in 3D dense prediction. Despite the promising results, there is still room for improvement. CUE/CUE+ cannot guarantee correctly estimated uncertainties for *all points* (see Fig. 1). We suppose that more sophisticated sampling strategies utilizing CES and CEM, such as using hardest triplet sampling or sampling among all points rather than points at objects' boundaries, would bring a better uncertainty estimation. This is an area of future research for us.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[2] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*, 2019, pp. 8958–8966.

[3] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2021, pp. 11 753–11 762.

[4] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*, 2019, pp. 6411–6420.

[5] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Proc. Eur. Conf. Comput. Vis*. Springer, 2020, pp. 681–699.

[6] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "Magface: A universal representation for face recognition and quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2021, pp. 14 225–14 234.

[7] M. Bucher, S. Herbin, and F. Jurie, "Improving semantic embedding consistency by metric learning for zero-shot classiffication," in *Proc. Eur. Conf. Comput. Vis*. Springer, 2016, pp. 730–746.

[8] S. Czolbe, K. Arnavaz, O. Krause, and A. Feragen, "Is segmentation uncertainty useful?" in *International Conference on Information Processing in Medical Imaging*. Springer, 2021, pp. 715–726.

[9] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Adv. Neural Inf. Process. Syst*, vol. 30, 2017.

[10] C. Qi, J. Yin, H. Liu, and J. Liu, "Neighborhood spatial aggregation based efficient uncertainty estimation for point cloud semantic segmentation," in *2021 IEEE Int. Conf. Robot. Auto.* IEEE, 2021, pp. 14 025–14 031.

[11] M. Monteiro, L. Le Folgoc, D. Coelho de Castro, N. Pawlowski, B. Marques, K. Kamnitsas, M. van der Wilk, and B. Glocker, "Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty," *Adv. Neural Inf. Process. Syst*, vol. 33, pp. 12 756–12 767, 2020.

[12] W. Li, X. Huang, J. Lu, J. Feng, and J. Zhou, "Learning probabilistic ordinal embeddings for uncertainty-aware regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2021, pp. 13 896–13 905.

[13] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*, 2021, pp. 7303–7313.

[14] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE Int. Conf. Robot. Auto.* IEEE, 2009, pp. 3212–3217.

[15] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2017, pp. 1802–1811.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2017, pp. 652–660.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Process. Syst*, vol. 30, 2017.

[19] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 3075–3084.

[20] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*, 2021, pp. 16 259–16 268.

[21] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*, 2019, pp. 6902–6911.

[22] J. Chang, Z. Lan, C. Cheng, and Y. Wei, "Data uncertainty learning in face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2020, pp. 5710–5719.

[23] F. Warburg, M. Jørgensen, J. Civera, and S. Hauberg, "Bayesian triplet loss: Uncertainty quantification in image retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*, 2021, pp. 12 158–12 168.

[24] Y. Zhang, C. Wang, and W. Deng, "Relative uncertainty learning for facial expression recognition," *Adv. Neural Inf. Process. Syst*, vol. 34, 2021.

[25] A. Jungo and M. Reyes, "Assessing reliability and challenges of uncertainty estimations for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 48–56.

[26] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[27] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.

[28] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *International Symposium on Visual Computing*. Springer, 2020, pp. 207–222.

[29] L. Tang, Y. Zhan, Z. Chen, B. Yu, and D. Tao, "Contrastive boundary learning for point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2022, pp. 8489–8499.

[30] M. Magdon-Ismail and J. T. Purnell, "Approximating the covariance matrix of gmms with low-rank perturbations," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2010, pp. 300–307.

[31] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 5545–5554.

[32] S. Horache, J.-E. Deschaud, and F. Goulette, "3d point cloud registration with multi-scale architecture and unsupervised transfer learning," in *Proc. Int. Conf. 3D Vis*. IEEE, 2021, pp. 1351–1361.

[33] C. Park, Y. Jeong, M. Cho, and J. Park, "Fast point transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2022, pp. 16 949–16 958.

[34] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2017, pp. 5828–5839.

[35] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 9621–9630.

[36] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2018, pp. 9224–9232.

[37] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE Int. Conf. Robot. Auto.* IEEE, 2016, pp. 4762–4769.