

Explanation of Uncertainty in Point Cloud Registration

Ziyuan Qin
Technical University of Munich
ziyuan.qin@tum.de

Abstract

Iterative Closest Point (ICP) is a commonly used algorithm to estimate transformation between two point clouds. However, robotic tasks such as grasping may fail due to large uncertainty in pose estimates. Thus, the uncertainty sources should be analyzed and attributed with importance. In this work, kernel SHAP is utilized to identify the importance of each uncertainty source in ICP, so that task execution can be successful after mitigating the sources. Experiment results show that this explanation method can reasonably explain uncertainty sources and be potentially used in practical scenarios. The code of the explanation module is available at <https://github.com/franklinqin0/explain-ICP-uncertainty>.

1 Introduction

Point cloud registration plays an essential role in many tasks in robotics and computer vision, such as simultaneous localization and mapping [1], grasping [2], and augmented reality [3]. Iterative Closest Point (ICP) is a widely used algorithm to register two point clouds [4]. Given the source and reference point clouds and an initial transformation estimate, ICP iteratively minimizes the Euclidean distance between pairs of matching points from both point clouds.

In practice, the ICP pose estimation process is usually affected by several sources of uncertainty. These include sensor noise, initial pose uncertainty, partial overlap, under-constrained situations, and intrinsic ICP randomness [5–7]. Due to the error incurred by these error sources, a single pose estimate is often insufficient to obtain robust transformation and accurately localize a moving object. Thus, many ICP algorithms provide not only the point estimate of transformation but also the uncertainty estimate for the transformation parameters [5–9]. Depending on how these sources of uncertainty vary, the uncertainty also varies.

Given the uncertainty of the pose estimate produced by any uncertainty-aware point cloud registration algorithm, this work employs kernel SHAP to build a model-agnostic explanation module and identify the association between uncertainty sources and the estimated uncertainty in point cloud registration [10], so that after removing or mitigating the sources, the pose estimates achieve a lower uncertainty and arrive at a better localization.

2 Background and Related Works

In this section, the background and related works of ICP, sources of uncertainty, and uncertainty estimation are presented.

2.1 Iterative Closest Point

ICP aligns a source point cloud to a target point cloud by estimating the rigid transformation (rotation and translation) between them [11]. Since the correct point correspondences are unknown and the searching process is not differentiable, it is generally impossible to determine the optimal rotation and translation in one step. Thus, iterations are performed, each consisting of two steps:

- Using an initial pose $\theta = \{x, y, z, \text{roll}, \text{pitch}, \text{yaw}\}$, the point correspondences are established by minimizing the distance function. There are two variants of distance function: point-to-point [4] and point-to-plane [12]. The commonly used point-to-point distance metric is:

$$\text{point-to-point}(s'_i, \mathcal{R}) = \min_{r_j \in \mathcal{R}} \|r_j - s'_i\|,$$

where $s_i, r_j \in \mathbb{R}^3$ are corresponding point pairs in 3D space belonging to source cloud $\mathcal{S} = \{s_i\}_{i=1}^N$ and reference cloud $\mathcal{R} = \{r_j\}_{j=1}^M$ respectively. $s'_i = (Rs_i + t)$ is a transformed point in the source cloud, $t \in \mathbb{R}^3$ is a translation vector consisting of $\theta_{1:3}$ and $R \in \mathbb{R}^{3 \times 3}$, parametrized by $\theta_{4:6}$, represents a rotation matrix.

- Estimate the relative rigid transformation R and t by minimizing the distance between each pair of corresponding points. The point-to-point cost function has the following form:

$$\operatorname{argmin}_{\theta} \mathcal{L}(\theta) = \frac{1}{N} \sum_i^N \| (R s_i + t) - r_j \|^2.$$

The estimated rigid transformation consists of rotation R and translation t . R is in special orthogonal group $\text{SO}(3) := \{R \in \mathbb{R}^{3 \times 3} \mid R^\top R = I, \det(R) = 1\}$ and $t \in \mathbb{R}^3$. Combining both rotation and translation, in homogeneous coordinate, the rigid transformation lives in the group of special Euclidean transformations: $\text{SE}(3) := \left\{ g = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \mid R \in \text{SO}(3), t \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4 \times 4}$.

2.2 Sources of ICP Uncertainty

According to the literature [5–7], the uncertainty of ICP uncertainty mainly comes from five possible sources: **sensor noise**, **initial pose uncertainty**, **partial overlap**, **under-constrained situations**, **intrinsic ICP randomness**.

- Sensor noise:** the sensor noise consists of both sensor white noise and sensor bias noise. Sensor white noise is caused when each point measured in a point cloud is influenced by an independent random sensor noise, a function of point depth and beam angle [13]. Sensor bias noise is induced when the observed points share common errors brought about by environmental conditions like temperature drift effect, distortion due to sensor calibration, and physical nature or texture of the perceived material [14].
- Initial pose uncertainty:** If initialization of pose has large uncertainty, the solution would converge to a local minimum rather than the attraction area of the true solution [7, 9].
- Partial overlap:** partial overlap refers to a situation where there is a limited amount of overlap between the points in the source and target point clouds. This limitation may arise from the viewpoint of the sensor, occlusions, or the relative motion between the sensor and the scene. It can be challenging to establish reliable correspondences between the two point clouds, leading to alignment uncertainty [6, 15].
- Under-constrained situations:** an ambiguous point cloud structure such as a rotational symmetric bottle would yield high uncertainty in rotation. This type of uncertainty can cause erroneous data associations and induce high pose estimation error [5].
- Intrinsic ICP randomness:** though the ICP process itself is deterministic, a certain degree of uncertainty is introduced by the random filtering process, e.g., sub-sampling and outlier rejection, such that even two solutions with exactly the same inputs might differ [16].

The last two sources, **under-constrained situations**^{*} and **intrinsic ICP randomness**, could easily be included in the kernel SHAP explanation module described in Section 3.5. However, for the sake of simplicity, they are not included in our attempt to explain and minimize ICP uncertainty. In this work, we focus on the first three sources of uncertainty: **sensor noise**, **initial pose uncertainty**, and **partial overlap**.

2.3 Uncertainty Estimation in ICP

There exist many ICP algorithms that either produce single transformation estimate or uncertainty estimates for the transformation parameters [6, 16]. Although uncertainty-based ICP algorithms are advantageous because they can provide a distribution of possible poses and estimate the pose uncertainty accordingly [5–7], the original ICP algorithm using point-to-point distance, proposed in [4], is chosen to demonstrate that the explanation method can indeed produce reasonable estimates. That is, if our explanation works for original ICP producing a single pose estimate, it could easily adapt to any other ICP algorithm, single estimate or not. Multiple pose estimates are sampled to compute uncertainty estimate using the Kullback-Leibler (KL) divergence. More details can be found in Section 5.2.

3 Explanation Methods

To attribute uncertainty to various sources, an explanation method is applied. The following talks about additive feature attribution method, LIME, Shapley values, and eventually how kernel SHAP combines these ideas into an linear model with desirable properties.

3.1 Additive Feature Attribution Method

Consider f as the initial model to be explained and g as the explanation model. Local techniques, as suggested in LIME, are employed to explain a prediction $f(x)$ using a single input x . Explanation models typically use *simplified* inputs x' that are transformed into the *original* inputs x by a mapping function $x = h_x(x')$. Local methods aim to ensure that $g(z') \approx f(h_x(z'))$ whenever $z' \approx x'$ [10].

Explanation Methods such as LIME and SHAP both belong to additive feature attribution method, which is based on a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i, \quad (1)$$

where $z' \in \{0, 1\}^M$, M is the number of simplified input features, and $\phi_i \in \mathbb{R}$.

Explanation models according to equation (1) assign an effect ϕ_i to each feature. By summing the effects of all feature attributions and bias ϕ_0 , one can approximate the output $f(x)$ of the original model.

3.2 LIME (Local Interpretable Model-agnostic Explanations)

In [17], a good explanation method can be identified with the following three criteria.

The first essential criterion is that they should be **interpretable**, i.e., offer a qualitative understanding between the input variables and the response. Although a simple model can perfectly explain itself and is easy to understand, to make the explanation of complex models easy to understand for most people, usually a small number of weighted features are provided to explain feature importance.

*For instance, when a cylindrical mug removes its handle, it becomes a rotationally symmetric shape, and the resulting uncertainty would inflate.

The second criterion is **local fidelity**. Fidelity shows how well the explanation approximates the prediction of the prediction model. While it is generally impossible for an explanation to be completely faithful unless it is the complete description of the model itself, it's imperative for an explanation to be at least *locally faithful*. That is, it should align with the model's behavior in the vicinity of the instance being predicted.

Lastly, an explainer should be able to explain *any* model, and thus be **model-agnostic**. This property allows the explanation method to explain not only models that are inherently interpretable, such as linear regression and decision tree, but also complex models like artificial neural networks.

As the name suggests, LIME incorporates the three criteria. As detailed in Section 3.5, Kernel SHAP also satisfies the aforementioned three properties and additional three desirable properties in Section 3.3.

The LIME explanation technique interprets individual model predictions by creating a local approximation of the model around a specific prediction [17]. LIME refers to simplified inputs as "interpretable inputs," and the transformation $x = h_x(x')$ converts a binary vector of interpretable inputs into the original input space. When LIME uses a local linear explanation model, it adheres to Equation (1), and is therefore an additive feature attribution method.

To find feature attributions ϕ , LIME minimizes the following objective function with respect to x :

$$\arg \min_{g \in G} L(f, g, \pi_{x'}) + \Omega(g) \quad (2)$$

where G is a set of potentially interpretable models, $L(f, g, \pi_{x'})$ is a measure of how unfaithful g is approximating f in the locality defined by $\pi_{x'}$, and $\Omega(g)$ quantifies complexity of explanation model, therefore encouraging interpretability. By minimizing the loss L across a set of samples in the simplified input domain weighted by the local kernel $\pi_{x'}$ and the complexity metric Ω , both local fidelity and interpretability are ensured.

3.3 Classic Shapley Values

In cooperative game theory, players cooperate in a coalition and receive a certain payoff. Depending on their individual contribution to the total payoff, **Shapley value** assigns contribution to each player. As shown in figure 1, in all pairs of coalitions that only differ on if member A is included, the marginal contribution of A is obtained by the taking the weighted average of the payoff differences. This idea in game theory can easily be extended to model explanations, where each player represents a feature, and the coalition value signifies model prediction. The methods using Shapley value to compute explanations of model predictions include Shapley regression values [18], Shapley sampling values [19], and Quantitative Input Influence [20].

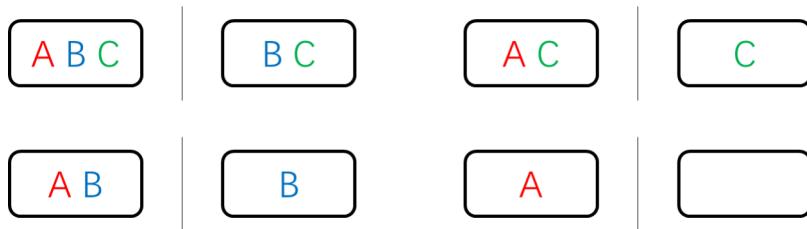


Figure 1: The importance of player A in coalition is determined by the weighted average of payoff differences in left coalitions subtracting those in the right ones.

There are three desirable properties with additive feature attribution methods: 1) local accuracy, 2) missingness, and 3) consistency. **Local accuracy** states that when $x = h_x(x')$, explanation model $g(x')$ matches the original model $f(x)$. **Missingness** requires that missing features where $x'_i = 0$ should have zero attribution: $\phi_i = 0$. **Consistency** indicates that if a

some simplified input's contribution increases or stays the same, the input's attribution should not decrease. While [21] demonstrated that local accuracy and consistency are already satisfied by the classical Shapley value estimation methods such as [18–20], missingness is required to adapt the Shapley values to the class of feature attribution methods.

Surprisingly, according to [10], there exists only one possible explanation model g that follows Equation (1) and satisfies the three properties:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \quad (3)$$

where M is the number of simplified input features, $|z'|$ is the number of non-zero entries in z' , and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x' .

In Equation (3), prediction $f_x(z')$ is made with the input feature i present, but $f_x(z' \setminus i)$ has the feature absent. The difference between these predictions $f_x(z') - f_x(z' \setminus i)$ represents the marginal contribution of feature i in the set z' . As the importance of a feature depends on other features in the model, the preceding differences are computed for all possible subsets $z' \subseteq x'$. The Shapley values ϕ_i is obtained by taking the weighted average of all possible differences.

For a given simplified input mapping h_x , Equation (3) is the only additive feature attribution method satisfying the three properties uses Shapley values. Methods not based on Shapley values can violate local accuracy or consistency.

3.4 SHAP (SHapley Additive exPlanation) Values

SHAP values are proposed by [10] as a unified measure of feature importance. They are Shapley values of the original model's conditional expectation function and thus the solution to Equation (3), where $f_x(z') = f(h_x(z')) = f(z_S) \approx E[f(z) | z_S]$, and S is the set of non-zero indices in z' . The simplified input mapping $z_S = h_x(z')$ has missing values for features not in the set S . When the model is non-linear or the input features are dependent, the order of adding features to the expectation matters. Because most models cannot handle arbitrary patterns of missing input values, $f(z_S)$ is approximated with $E[f(z) | z_S]$.

SHAP values provide the unique additive feature importance measure that adheres to the three properties and uses conditional expectations to define simplified inputs. While precise calculation of SHAP values can be computationally expensive, the subsequent Kernel SHAP method achieves similar approximation accuracy with fewer evaluations of the original model.

3.5 Kernel SHAP (Linear LIME + Shapley values)

Linear LIME employs a linear explanation model to locally approximate f in the simplified binary input space. Since linear LIME is an additive feature attribution method and Shapley values are the only possible solution to Equation (2) that satisfies the three properties – local accuracy, missingness, and consistency. Whether the solution to Equation (2) recovers Shapley values depends on the choice of loss function L , weighting kernel $\pi_{x'}$, and regularization term Ω . The LIME choices for these parameters are made heuristically and does not retrieve Shapley values. However, **Shapley kernel**, proposed and proved in [10], can induce Shapley values consistent with Equation (1) and the three properties to solve Equation (2):

$$\Omega(g) = 0 \quad (4a)$$

$$\pi_{x'}(z') = \frac{(M - 1)}{\binom{M}{|z'|}|z'|!(M - |z'|)} \quad (4b)$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} (f(h_x(z')) - g(z'))^2 \pi_{x'}(z') \quad (4c)$$

Given that $g(z')$ is assumed to follow a linear form in Equation (1) and L represents a quadratic loss, Equation (2) can be solved via *weighted linear regression*. The weighting function $\pi_{x'}$ forms the foundation of the Shapley kernel, and the entire process of weighted linear regression is termed as **kernel SHAP**.

Because kernel SHAP follows linear LIME, it satisfies the three criteria of LIME: interpretable, local fidelity, and model-agnostic. Because it is a specific implementation of Shapley values, it complies with the three desirable properties: local accuracy, missingness, and consistency.

4 Method

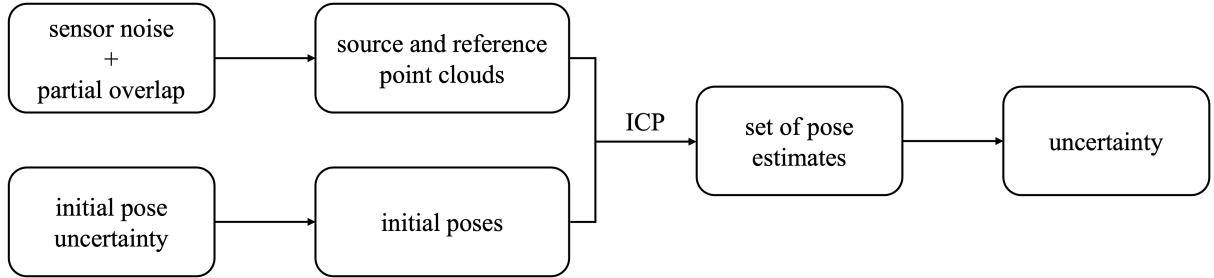


Figure 2: How the uncertainty is impacted by uncertainty sources.

As Figure 2 shows, sensor noise and partial overlap affect the source and reference point clouds, whereas initial pose uncertainty influences the range of sampled initial poses. As mentioned in Section 2.1 and 2.3, ICP takes the input point clouds and initial poses and produces a set of pose estimates. Finally, uncertainty is estimated using KL divergence in Section 5.2. Given different levels of perturbation, the input point clouds and initial poses would change, and the ICP algorithm would yield different uncertainty estimates. Kernel SHAP, as described in Section 3.5, is then used to explain how important each uncertainty source contributes to the uncertainty estimate. As kernel SHAP uses the additive feature attribution method in Equation (1), the explanation is:

$$g(z') = \phi_{sn}z'_{sn} + \phi_{ip}z'_{ip} + \phi_{po}z'_{po} \quad (5)$$

where $z'_{sn}, z'_{ip}, z'_{po} \in \{0, 1\}$ represent simplified input features, $\phi_{sn}, \phi_{ip}, \phi_{po} \in \mathbb{R}$ refer to the attributed importance of initial pose uncertainty, sensor noise, and partial overlap, respectively. For an uncertainty source i , $z'_i = 1$ signifies that the feature is present, and input is perturbed in this fashion, whereas 0 indicates the absence of uncertainty source i , thus not perturbed by such feature and is thus the reference value. Note that bias ϕ_0 is omitted, as the expected value for the unperturbed case is 0, i.e., when z' are all 0.

Algorithm 1 Kernel SHAP estimates feature contribution for one prediction

Require: f : ICP algorithm that estimates uncertainty, $x = \{x_{sn}, x_{ip}, x_{po}\}$: instance to explain, M : number of features is 3, r : reference values are $\{0, 1, 0\}$ (see details in Section 5.1)

for each coalition $z'[i, j] \in \{0, 1\}$, $i \in \{1, \dots, 2^M\}$, $j \in \{1, \dots, M\}$ **do**

- $z[i] \leftarrow h_x(z'[i])$ ▷ $z'[i, j] = 1$ maps $z[i, j]$ to $x[j]$, and 0 maps $z[i, j]$ to $r[j]$
- $y[i] \leftarrow f(z[i])$ ▷ Calculate estimated uncertainty with ICP algorithm
- $w[i] \leftarrow \pi_{x'}(z'[i])$ ▷ Calculate weights by Equation 4b

end for

$\phi = (z'^T w x)^{-1} z'^T w y$ ▷ closed-form solution for weighted linear regression

return ϕ

Instance x	$x' =$	sensor noise initial pose partial overlap			$x =$	sensor noise initial pose partial overlap		
		1	1	1		0.09	1.1	0.09
Instance with "absent" features	$z' =$	sensor noise	initial pose	partial overlap	$z =$	sensor noise	initial pose	partial overlap
		1	0	0		0.09	1.1	0.09
							1.0	0

Figure 3: The simplified input on the left, represented by binary conditions, are mapped to the feature values on the right via the mapping function h_x .

In Algorithm 1, given a perturbation setting $x = \{x_{sn}, x_{ip}, x_{po}\}$ and $M = 3$, for each i in $2^M = 8$, $z'[i]$ is mapped to $z[i]$ via h_x , as shown in Figure 3. Then, uncertainty $y[i]$ is predicted by the ICP algorithm and weight $w[i]$ are calculated by Equation (4b). Finally, the quadratic loss L in Equation (4c) is minimized by the optimal SHAP values ϕ calculated via the closed form solution for weighted linear regression.

In the following, the three ways to perturb two input point clouds and the initial pose estimate are presented.

4.1 Sensor Noise

As described in Section 2.2, sensor noise is comprised of sensor white noise and sensor bias noise. Since the sensor white noise is more straightforward to model, it is represented as a zero-mean Gaussian noise without loss of generality. The noise is then added to each point in the two input point clouds.

4.2 Initial Pose Uncertainty

As derived in [7], given a small rotation α, β, γ around the x, y, z axes respectively, the full rotation R can be linearly approximated as:

$$R \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} = I_3 + \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = I_3 + \delta^\wedge$$

where the operator \wedge turns the vector $\delta = [\alpha \ \beta \ \gamma]^T$ in \mathbb{R}^3 to a 3×3 skew-symmetric matrix in Lie algebra $so(3)$, the tangent space to Lie group $SO(3)$.

Combined with translation ρ , the rigid transformation can be linearized as $T \approx I_4 + \xi^\wedge$ with

$$\xi^\wedge := \begin{bmatrix} \delta^\wedge & \rho \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \xi = \begin{bmatrix} \delta \\ \rho \end{bmatrix}$$

where $\delta \in \mathbb{R}^3$ represents rotation, $\rho \in \mathbb{R}^3$ signifies translation, and operator \wedge turns $\xi \in \mathbb{R}^6$ to a 4×4 real matrix in Lie algebra $se(3)$, the tangent space to Lie group $SE(3)$.

If ξ is taken randomly as a Gaussian $\xi \sim \mathcal{N}(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{6 \times 6}$ is the covariance matrix, then $I_4 + \xi^\wedge$ defines a small transformation. For the ground truth pose T_{gt} , the initial guess is a randomly perturbed pose close to T_{gt} : $T_{init} = T_{gt}(I_4 + \xi^\wedge) = T_{gt} + T_{gt}\xi^\wedge$. Using the notion of concentrated Gaussian distribution [22],

$$T_{init} = T_{gt} \exp(\xi^\wedge) \quad (6)$$

where $\xi \sim \mathcal{N}(0, \Sigma) \in \mathbb{R}^6$, and the $\exp(\cdot)$ operator denotes the exponential map from $\text{se}(3)$ to $\text{SE}(3)$.

Note that ξ can be viewed as the error between T_{gt} and T_{init} . Indeed, the relative transformation between the ground truth and initial poses is encoded in ξ as $\exp(\xi^\wedge) = T_{gt}^{-1}T_{init}$.

4.3 Partial Overlap

For a pair of input point clouds P_1^l and P_2^l , an overlapping ratio can be computed similar to that in [15]. Given ground truth absolute pose T_1 and T_2 , the point clouds are transformed from local camera frame to world frame: P_1 and P_2 . For each point p in P_2 , the nearest neighbor $NN(p)$ in P_1 is found using kNN with $k = 1$. The neighbor is valid only if the Euclidean distance between p and $NN(p)$ is no larger than $d = 0.2$. Calculating the ratio between the number of valid neighbors and number of points in P_1 yields the overlap ratio:

$$O_{1,2} = \frac{N}{|P_1|}, \text{ where } N = \sum_{p \in P_2} \mathbb{1}(\|NN(p) - p\| \leq d).$$

In reality, the means to perturb input point clouds P_1 and P_2 with partial overlap are various and depend on the specific scenario, a simple but effective way to reduce overlapping region is as follows. Given the current overlap ratio $O_{1,2}$, P_2 is perturbed to reach target overlap ratio $O'_{1,2} = O_{1,2} - \lambda$, with $\lambda > 0$. By simple calculation, $N - |P_1| \cdot O'_{1,2}$ points need to be removed from the overlapping region in P_2 to reduce current overlap ratio by λ .

5 Uncertainty Sources and Estimation

After learning how to apply kernel SHAP and perturb input in section 4, this section introduces the perturbation domain and uncertainty estimation.

5.1 Domains of Uncertainty Sources

To analyze the effect of each uncertainty source: sensor noise, initial pose uncertainty, and partial overlap, a sensible range for each source is specified.

As the mean bias value for Hokuyo sensor is found as 5 cm in [14], the Gaussian *sensor noise* $\mathcal{N}(0, \sigma)$ is added to the two input point clouds with σ from 0 cm to 10 cm, with a step size of 1 cm.

As shown in Equation (6), T_{init} is perturbed by the small *initial pose uncertainty* $\xi \sim \mathcal{N}(0, s\Sigma)$, which is applied with a scale s from 1 to 2, with a step size of 0.1.

Since the number of points in each point cloud is on the order of 10^5 , removing points in the *partial overlap* region of two point clouds would result in a big change in uncertainty estimate. Thus, λ , difference between original overlap ratio $O_{1,2}$ and target overlap ratio $O'_{1,2}$, is set relatively small. Given $O_{1,2} > 0.1$, λ ranges from 0 to 0.1, with a step size of 0.01.

5.2 Uncertainty Estimation

After applying the three sources of uncertainty to perturb input point clouds and initial pose in different degrees, uncertainty of the estimated pose T_{pert} should be estimated in the form of a scalar. In this work, we utilize the KL divergence between a pseudo-true distribution and the estimated perturbed distribution. The pseudo-true distribution is computed by sampling 100 ICP pose estimates of unperturbed inputs: initial pose sampled with zero-mean Gaussian with $s\Sigma$, where $s = 1$, around the ground truth pose, without sensor noise, and unaltered overlap ratio. In this way, the estimated uncertainty is the combined influence of all three uncertainty sources.

6 Experimental Results

This section shows how the explanation method kernel SHAP, as illustrated in Section 3.5, attributes importance to each of the three uncertainty sources for a pair of point clouds. As described in Section 4, the three uncertainty sources would in general cause the uncertainty to become larger in different ways and degrees. When they are all absent, the unperturbed uncertainty estimated using KL divergence detailed in Section 5.2 would yield 0. Thus, it makes sense for the attribution values ϕ to the uncertainty sources to be positive for the perturbed cases and zero for the unperturbed cases.

6.1 Different Perturbations for the Same Pair of Input Point Clouds in *Apartment*

In this section, we examine various perturbation sets to understand how kernel SHAP elucidates the sources of uncertainty in point clouds 6 and 7 from the *Apartment* sequence in the *Challenging datasets* [23]. The domains of these perturbations are detailed in Section 5.1. We employ summary plots to demonstrate the overall impact of each source of uncertainty. Additionally, feature dependence plots are used to illustrate both the relationship between the magnitude of an uncertainty source and its SHAP value, and the interactions between a current feature and other sources of uncertainty. Finally, waterfall plots are presented to show the contribution of each uncertainty source to individual data instances.

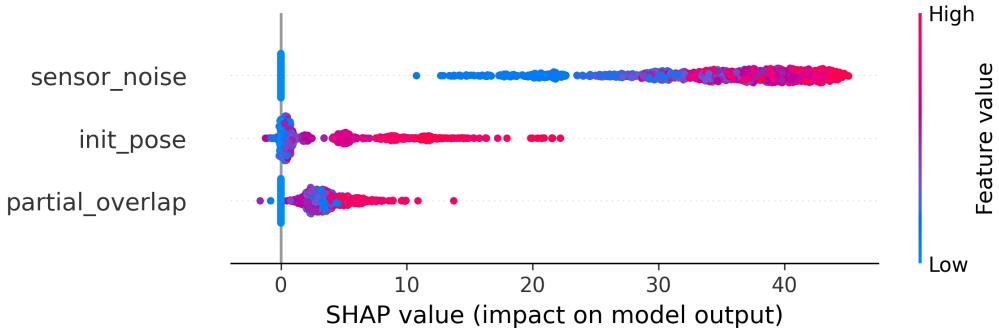


Figure 4: SHAP summary plot.

6.1.1 Summary Plot

In the summary plot illustrated in Figure 4, sensor noise correspond to the largest SHAP values, indicating that they contribute the most to uncertainty for input point clouds 6 and 7 in *Apartment*. Most of the SHAP values are positive, and some are zero, just as predicted, meaning these three uncertainty sources, if present, increase uncertainty. However, there are a few cases where the SHAP values are negative, as opposed to our predictions. These points can also be seen in feature dependence plots in Figure 5.

6.1.2 Feature Dependence Plot

In the dependence plots shown in Figure 5, both sensor noise and initial pose uncertainty show a positive correlation with respect to SHAP value. For partial overlap, the small overlap ratio results in relatively low SHAP value, in line with the expectation, but the general trend is nonlinear. This may happen because the points removal process is random when perturbing with partial overlap, leading to counterintuitive SHAP values.

The feature dependence plots are shown in Figure 5. In Figures 5a and 5b, SHAP values of sensor noise are shown with feature values of initial pose uncertainty and partial overlap,

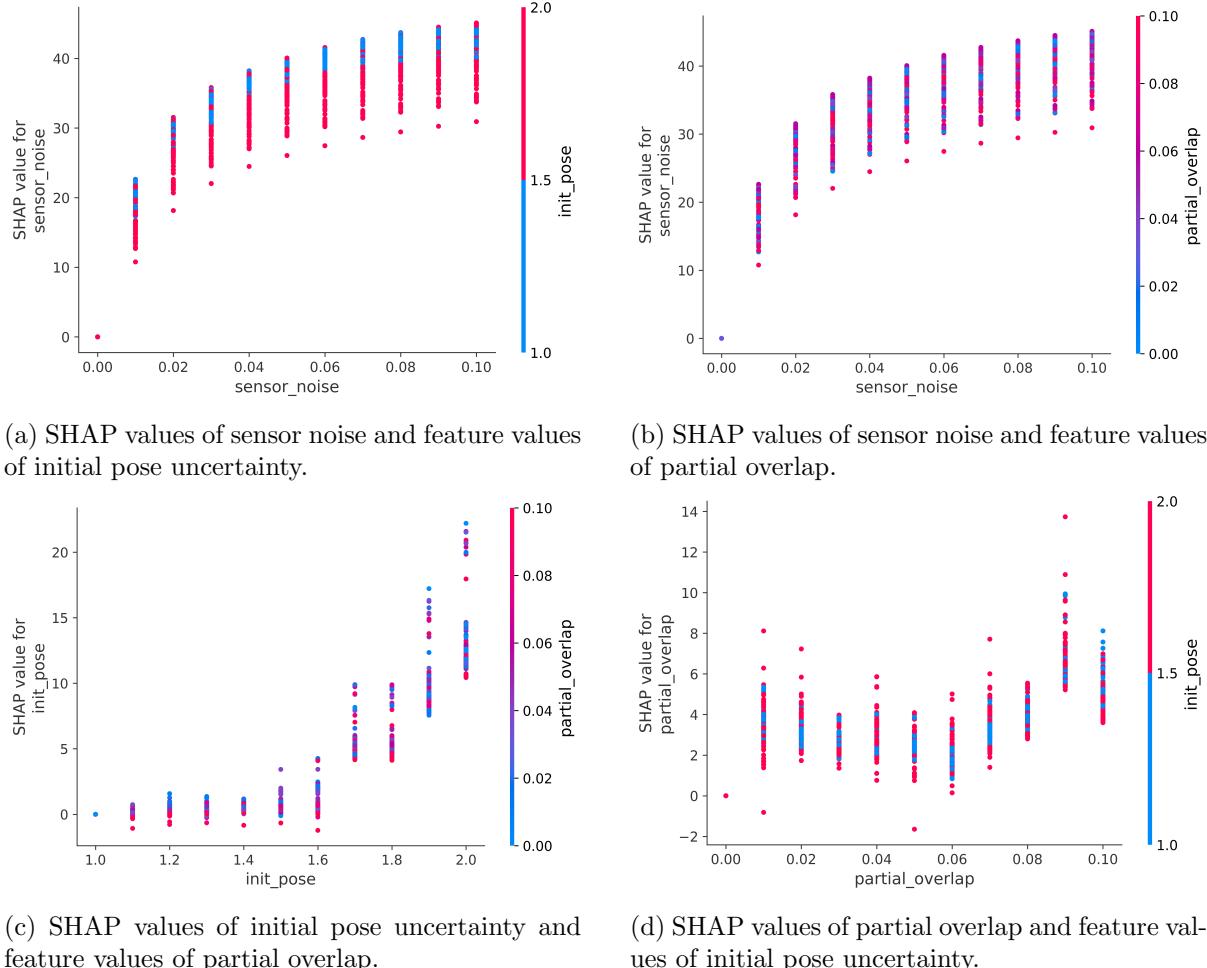
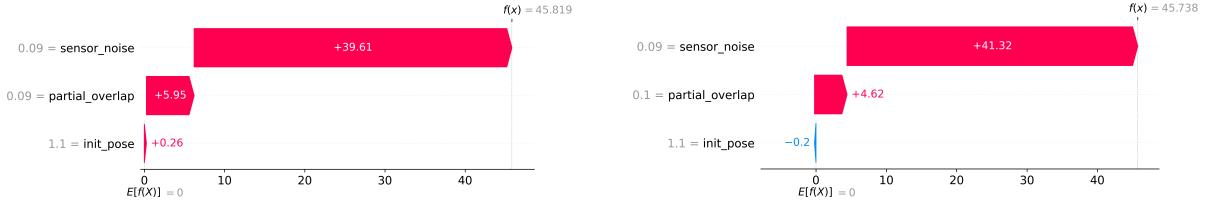


Figure 5: SHAP dependence plots.

respectively. In both plots, SHAP values of sensor noise are low when the interaction feature has strong effect. In Figure 5c, SHAP values are sometimes negative when the partial overlap is strong, the case illustrated by Figure 6b. Similarly, Figure 5d also showcases negative SHAP values for partial overlap when initial pose uncertainty is stark. Since these three features are not independent, the presence of one uncertainty source can reduce the importance of another source, sometimes resulting in negative SHAP values.

6.1.3 Waterfall Plot

The Figure 6 presents two waterfall plots, analyzing the feature importance for two similar data instances. In Figure 6a, sensor noise is 9 cm, initial pose uncertainty has scale 1.1, and partial overlap eliminates 9% points in overlapping region of reference point cloud 7. Among the three uncertainty sources, sensor noise is more important than partial overlap, and initial pose uncertainty has the least influence on uncertainty. Figure 6b, however, is one of the few abnormal cases shown in Figure 4 and 5c. Here, partial overlap increases to the maximum 10%, but initial pose uncertainty assumes a negative SHAP value, -0.2. Such anomaly, as explained in Section 6.1.2, may be caused by feature dependence.



(a) SHAP values for $\{0.09, 1.1, 0.09\}$.

(b) SHAP value for $\{0.09, 1.1, 0.1\}$.

Figure 6: SHAP waterfall plots.

6.2 Same Perturbation for Contiguous Pairs of Point Clouds in Different Sequences

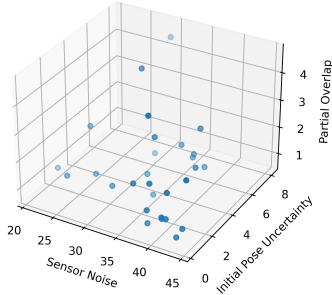
In this section, the perturbation on the three uncertainty sources is fixed to the mean value: $\{0.05, 1.5, 0.05\}$. The feature effects are now analyzed on every contiguous pairs of point clouds in all 8 sequences of *Challenging datasets*. This experimental setting is more aligned with practical use cases, where the perturbation is set, and after summarizing the overall feature effects, the importance of each uncertainty sources is determined.

After eliminating outliers using interquartile range, the SHAP values of the three uncertainty sources are shown in Figure 7. Although the perturbation is the same on all sequences, the effect seems to be most prominent in *Apartment*, shown from the larger ranges in all three dimensions.

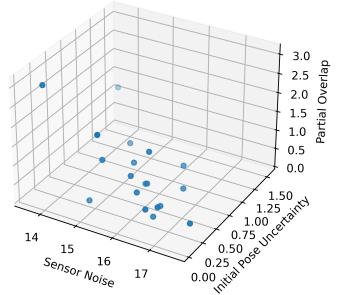
Table 1 presents the median SHAP values of each uncertainty source in all sequences. It could be clearly seen that except for the sequence *Mountain*, all other sequences have the order of SHAP values **sensor noise > partial overlap > initial pose uncertainty**. Thus, we could safely conclude that in such perturbation setting, eliminating sensor noise would reduce uncertainty on all sequences considerably.

Sequence	Sensor Noise	Initial Pose Uncertainty	Partial Overlap
Apartment	35.571175	1.415812	2.478877
ETH	15.392886	0.278460	1.279239
Stairs	11.755226	0.287785	0.492705
Mountain	3.199889	1.055795	0.973968
Gazebo Summer	12.524350	0.380334	4.750572
Gazebo Winter	16.923138	2.461525	3.904520
Wood Summer	11.811761	1.138083	3.617080
Wood Autumn	12.268561	1.345611	2.197536

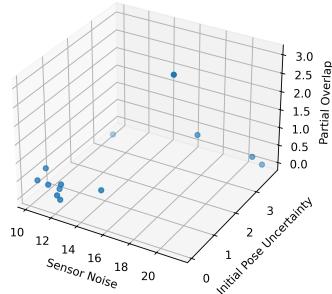
Table 1: Median SHAP values of uncertainty sources in all sequences. Except for the sequence *Mountain*, in every other sequence, **partial overlap** has more importance than **initial pose uncertainty**.



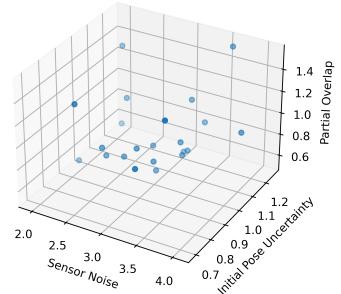
(a) Apartment



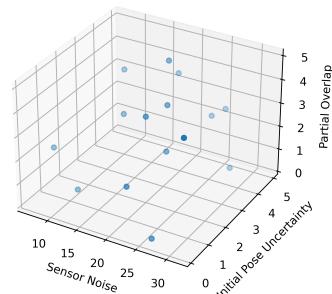
(b) ETH



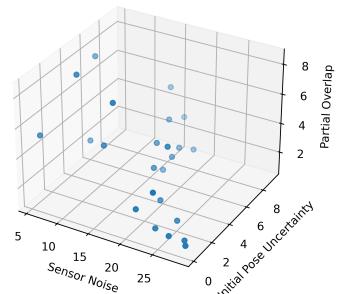
(c) Stair



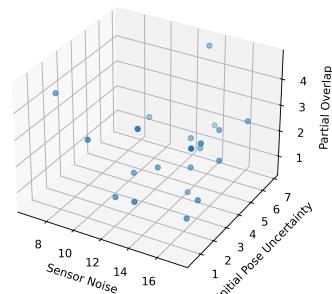
(d) Mountain



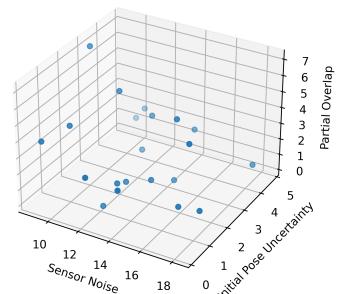
(e) Gazebo Summer



(f) Gazebo Winter



(g) Wood Summer



(h) Wood Autumn

Figure 7: Inlier SHAP values of three uncertainty sources in all sequences.

7 Conclusion

In this work, kernel SHAP is employed to explain the association between uncertainty sources and uncertainty estimate. In experiment implemented in Section 6.1, the effects of each uncertainty source under different perturbation levels are examined for the same input point clouds. Although there exist nonlinear trend for partial overlap vs. SHAP values and negative SHAP values for initial pose uncertainty and partial overlap, most SHAP values are nonnegative and make sensible explanations. In the experiment in Section 6.2, the effects of uncertainty sources under the same perturbation level are scrutinized for contiguous pairs of point clouds in all sequences. This scenario is more aligned with practical use cases, and could provide insights on the importance of each uncertainty source.

As kernel SHAP is model-agnostic, it could explain other ICP algorithms, such as Stein ICP [6]. If the uncertainty sources change their range or stepwise, or some other uncertainty sources are also considered, e.g., under-constrained situations and ICP randomness, kernel SHAP would still be able to explain.

8 Caveats

Granted, kernel SHAP could effectively explain the effect of each uncertainty source for one ICP uncertainty estimate, there exist some *caveats* for the three chosen uncertainty sources and explanation methods.

- In the original data, sensor noise is assumed to be zero for unperturbed data, but this may be an oversimplified assumption. In reality, the input point clouds would be subjected to sensor noise and current sensor noise perturbation level should be quantified. To attain the reference value of 0, they would have to be denoised.
- The setting of experiments in this work follows from [7], where initial poses are sampled around ground truth pose. But in practice, this ground truth pose may not be known, and the initial poses may have to be uniformly sampled from a range obtained from some coarse alignment algorithms.
- When overlap ratio is already very small, e.g., around 0.4, removing even 10% of points in overlapping region may induce unreasonable SHAP values.
- Kernel SHAP could identify how important a feature is to the model, but if the model changes, uncertainty has to be recalculated and conclusions of feature importance might differ.
- As explanation methods trade interpretability with complexity, they may be able to simplify prediction of one data instance to a linear explanation model, but cannot capture the complexity of the original model. This is an inherent limitation in interpretable explanation models.

9 Future Work

Though explanation methods could evaluate the correlation between features and prediction, as correlation does not imply causation, the causal relationship between sources and estimate still remains unknown and requires knowledge of causal inference. This topic, if possible, will be for future research.

References

- [1] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.
- [2] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, “Pointnet-gpd: Detecting grasp configurations from point sets,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.
- [3] M.-L. Wu, J.-C. Chien, C.-T. Wu, and J.-D. Lee, “An augmented reality system using improved-iterative closest point algorithm for on-patient medical image visualization,” *Sensors (Basel, Switzerland)*, vol. 18, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:51909358>
- [4] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.
- [5] A. Censi, “An accurate closed-form estimate of icp’s covariance,” in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3167–3172.
- [6] F. A. Maken, F. Ramos, and L. Ott, “Stein icp for uncertainty estimation in point cloud matching,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1063–1070, 2021.
- [7] M. Brossard, S. Bonnabel, and A. Barrau, “A new approach to 3d icp covariance estimation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 744–751, 2020.
- [8] F. A. Maken, F. Ramos, and L. Ott, “Estimating motion uncertainty with bayesian icp,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8602–8608.
- [9] D. Landry, F. Pomerleau, and P. Giguere, “Cello-3d: Estimating the covariance of icp in the real world,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8190–8196.
- [10] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in neural information processing systems*, vol. 30, 2017.
- [11] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE, 2001, pp. 145–152.
- [12] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [13] Z. Wang, Y. Liu, Q. Liao, H. Ye, M. Liu, and L. Wang, “Characterization of a rs-lidar for 3d perception,” in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2018, pp. 564–569.
- [14] F. Pomerleau, A. Breitenmoser, M. Liu, F. Colas, and R. Siegwart, “Noise characterization of depth sensors for surface inspections,” in *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*. IEEE, 2012, pp. 16–21.
- [15] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss, “OverlapNet: Loop Closing for LiDAR-based SLAM,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

- [16] F. Pomerleau, F. Colas, R. Siegwart *et al.*, “A review of point cloud registration algorithms for mobile robotics,” *Foundations and Trends® in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.
- [17] M. T. Ribeiro, S. Singh, and C. Guestrin, ““why should i trust you?” explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.
- [18] S. Lipovetsky and M. Conklin, “Analysis of regression in game theory approach,” *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.
- [19] E. Štrumbelj and I. Kononenko, “Explaining prediction models and individual predictions with feature contributions,” *Knowledge and information systems*, vol. 41, pp. 647–665, 2014.
- [20] A. Datta, S. Sen, and Y. Zick, “Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems,” in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 598–617.
- [21] H. P. Young, “Monotonic solutions of cooperative games,” *International Journal of Game Theory*, vol. 14, pp. 65–72, 1985.
- [22] G. Bourmaud, R. Mégrét, M. Arnaudon, and A. Giremus, “Continuous-discrete extended kalman filter on matrix lie groups using concentrated gaussian distributions,” *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 209–228, 2015.
- [23] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, “Challenging data sets for point cloud registration algorithms,” *The International Journal of Robotics Research*, vol. 31, no. 14, pp. 1705–1711, Dec. 2012.