Appendix of Kernel-SDF: An Open-Source Library for Real-Time Signed Distance Function Estimation using Kernel Regression

Zhirui Dai[1,*] Tianxing Fan[1,*] Mani Amani[1] Jaemin Seo[2]
Ki Myung Brian Lee[1] Hyondong Oh[2] Nikolay Atanasov[1]

This appendix is provided as a separate document with our code release. To obtain the latest information, please refer to our GitHub repository: https://github.com/ExistentialRobotics/kernel_sdf.

I. EM Algorithm for Updating BHM

Readers can refer to [1], [2] for more details about the Bayesian Hilbert Map (BHM). Here, we provide a brief derivation of the EM algorithm for updating the BHM model parameters, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

BHM represents a continuous occupancy field $P(y \mid \mathbf{x})$ by using a set of M hinge points $\tilde{\mathbf{X}}$ with weights $\mathbf{w}$ to do kernel regression in the Hilbert space:

$$P(y \mid \mathbf{x}, \mathbf{w}) = \sigma\big((2y-1)\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})\big), \tag{1}$$

where $\boldsymbol{\phi}(\mathbf{x}) = [k(\mathbf{x}, \tilde{\mathbf{x}}_1), \cdots, k(\mathbf{x}, \tilde{\mathbf{x}}_M), 1]^\top$ is a vector of kernel values, $k(\cdot,\cdot)$ is RBF kernel, and σ is the sigmoid function.

Given a dataset $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^K$ of K generated samples at time t, our goal is to update the weights $\mathbf{w}$ of the model such that we get the posterior distribution:

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}, \tag{2}$$

where $P(\mathbf{w})$ is the prior distribution of the weights, and the likelihood is given by:

$$\begin{aligned} P(\mathcal{D}|\mathbf{w}) &= \prod_{k=1}^K P(y_k|\mathbf{x}_k, \mathbf{w}) \\ &= \prod_{k=1}^K \sigma(-y_k \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_k)), \quad y_k \in \{-1, 1\}. \end{aligned} \tag{3}$$

Manuscript received: March 30, 2026; Revised June 17, 2026; Accepted July 6, 2026. This paper was recommended for publication by Editor Ayoung Kim upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by ARL DCIST CRA W911NF-17-2-0181, NSF FRR CAREER 2045945, the Ministry of Trade, Industry, and Energy (MOTIE), Korea, under the Strategic Technology Development Program supervised by the Korea Institute for Advancement of Technology (KIAT) [Grant No. P0026052], and Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, Forestry(IPET) through (Smart Farm Innovation Technology Development Program), funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) (RS-2025-02219411).

[1]Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA.

[2]Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34051, Republic of Korea. Emails: {zhdai,t2fan,mamai5250,kmblee,natanasov}@ucsd.edu; {jam.seo,h.oh}@kaist.ac.kr.

*Equal contribution. Code and additional results: https://github.com/ExistentialRobotics/kernel_sdf.

The marginal likelihood $P(\mathcal{D})$ is intractable, so we resort to approximate inference methods. Our goal to maximize the log marginal likelihood:

$$\begin{aligned} \log P(\mathcal{D}) &= \log \int P(\mathcal{D}|\mathbf{w})P(\mathbf{w})d\mathbf{w} \\ &= \log \int Q(\mathbf{w}) \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{Q(\mathbf{w})} d\mathbf{w} \\ &\geq \int Q(\mathbf{w}) \log \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{Q(\mathbf{w})} d\mathbf{w}, \\ &\uparrow \text{Jensen's inequality} \end{aligned} \tag{4}$$

where $Q(\mathbf{w})$ is an approximate posterior distribution of the weights.

Since $P(\mathcal{D}|\mathbf{w})$ is product of sigmoids that contains the weights $\mathbf{w}$ inside, we use the variational bound [1] to get a looser lower bound:

$$\sigma(r) \geq \sigma(\xi) \exp\left(\frac{r-\xi}{2} + \lambda(\xi)\left(r^2 - \xi^2\right)\right),$$

where ξ is a variational parameter and $\lambda(\xi) = \frac{1}{2\xi}\left(\frac{1}{2} - \sigma(\xi)\right)$.

Using the idea of Sequential BHM EM update [2], we assume a Gaussian prior $P(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ and a Gaussian approximate posterior $Q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ for the weights. Hence, we can write the variational lower bound of the log marginal likelihood as:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_Q\left[\log P(\mathcal{D}|\mathbf{w})\right] + \mathbb{E}_Q\left[\log P(\mathbf{w})\right] - \mathbb{E}_Q\left[\log Q(\mathbf{w})\right] \\ &\geq \sum_{k=1}^K \mathbb{E}_Q\left[\log \sigma(-y_k \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_k))\right] \\ &\quad + \mathbb{E}_Q\left[\log P(\mathbf{w})\right] - \mathbb{E}_Q\left[\log Q(\mathbf{w})\right] \\ &\geq \sum_{k=1}^K \left(\log \sigma(\xi_k) - \frac{\xi_k}{2} + \frac{\mathbb{E}_Q[r_k]}{2} + \lambda(\xi_k)\left(\mathbb{E}_Q\left[r_k^2\right] - \xi_k^2\right)\right) \\ &\quad + \mathbb{E}_Q\left[\log P(\mathbf{w})\right] - \mathbb{E}_Q\left[\log Q(\mathbf{w})\right] \\ &= \sum_{k=1}^K \left(\log \sigma(\xi_k) - \frac{\xi_k}{2} - \xi_k^2 \lambda(\xi_k) + (y_k - \frac{1}{2})\boldsymbol{\mu}_t^\top \boldsymbol{\phi}(\mathbf{x}_k)\right) \\ &\quad + \sum_{k=1}^K \left(\lambda(\xi_k)\boldsymbol{\phi}^\top(\mathbf{x}_k)\left(\boldsymbol{\Sigma}_t + \boldsymbol{\mu}_t\boldsymbol{\mu}_t^\top\right)\boldsymbol{\phi}(\mathbf{x}_k)\right) \\ &\quad + \frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_{t-1}^{-1}\right)\boldsymbol{\Sigma}_t\right) + \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_t|}{|\boldsymbol{\Sigma}_{t-1}|} \\ &\quad - \frac{1}{2}\left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\right)^\top \boldsymbol{\Sigma}_{t-1}^{-1}\left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\right), \end{aligned}$$

where $r_k = y_k \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_k), y_k \in \{0, 1\}$.

To maximize the lower bound $\mathcal{L}$, we take derivatives with respect to the variational parameters $\{\xi_k\}_{k=1}^K$ and $\boldsymbol{\mu}_t$, and set them to zero. This results in the following update equations:

a) E-Step: $\partial\mathcal{L}/\partial\boldsymbol{\mu}_t = 0$ leads to

$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t \left(\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\mu}_{t-1} + \sum_{k=1}^{K} \left(y_k - \frac{1}{2} \right) \boldsymbol{\phi}(\mathbf{x}_k) \right), \quad (5)$$

$$\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Sigma}_{t-1}^{-1} + \sum_{k=1}^{K} \left| \frac{1/2 - \sigma(\xi_k)}{\xi_k} \right| \boldsymbol{\phi}(\mathbf{x}_k)\boldsymbol{\phi}^\top(\mathbf{x}_k). \quad (6)$$

b) M-Step: $\partial\mathcal{L}/\partial\xi_k = 0$ leads to

$$\xi_k = \sqrt{\boldsymbol{\phi}^\top(\mathbf{x}_k) \left(\boldsymbol{\Sigma}_t + \boldsymbol{\mu}_t\boldsymbol{\mu}_t^\top \right) \boldsymbol{\phi}(\mathbf{x}_k)}, \quad \forall k = 1, \ldots, K. \quad (7)$$

Note that $\xi_{k,t} = \xi_k$, where t is omitted for simplicity. We suggest initialize ξ_k to 0.0 or 1.0 for all k (Empirically, both work well).

A. Practical Optimization for the Implementation

For each generated dataset $\mathcal{D}$, usually 1 to 2 EM iterations are sufficient for convergence. To make the algorithm more efficient, we set $\phi_m(\mathbf{x}_k) = 0$ if its value is smaller than a threshold (e.g., 10^{-3}) so that we can speed up the computation by the sparsity of $\boldsymbol{\phi}(\mathbf{x}_k)$. Besides, the update of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ needs matrix inversion of $\boldsymbol{\Sigma}_t^{-1}$, which is handled by Cholesky decomposition for efficiency and numerical stability.

II. Approximation of BHM Prediction

A. Mean Occupied Probability

With enough iterations, we get the converged $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and predict the occupancy probability at $\mathbf{x}_*$ by (1):

$$g(\mathbf{x}_*, \mathbf{w}) = P(y = 1 \mid \mathbf{x}_*, \mathbf{w}) = \sigma(\mathbf{w}^\top \boldsymbol{\phi}_*), \quad (8)$$

where $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\phi}_* = \boldsymbol{\phi}(\mathbf{x}_*)$. Let $g_*(\mathbf{w}) = g(\mathbf{x}_*, \mathbf{w})$. Then, the mean of $g_*(\mathbf{w})$ is

$$\mathbb{E}[g_*(\mathbf{w})] = P(y = 1 \mid \mathbf{x}_*) = \int \sigma\left(\mathbf{w}^\top \boldsymbol{\phi}_*\right) P(\mathbf{w}) d\mathbf{w}. \quad (9)$$

However, we cannot compute the above integral analytically. One way of approximation is to collect N samples of $\mathbf{w}$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, so that

$$\mathbb{E}[g_*(\mathbf{w})] = P(y = 1 \mid \mathbf{x}_*) \approx \frac{1}{N} \sum_{i=1}^{N} \sigma(\mathbf{w}_i^\top \boldsymbol{\phi}_*). \quad (10)$$

A faster and more accurate approximation is

$$\mathbb{E}[g_*(\mathbf{w})] \approx \sigma\left(h\right), \quad h = \frac{\boldsymbol{\mu}^\top \boldsymbol{\phi}_*}{\sqrt{1 + \frac{\pi}{8}\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}}. \quad (11)$$

Proof. We approximate the sigmoid function $\sigma(x)$ with the cumulative distribution function of the normal distribution, $\Phi(\lambda x)$, where $\lambda = \sqrt{\pi/8}$. Therefore,

$$\begin{aligned}
&P(y = 1 \mid \mathbf{x}_*) \\
&\approx \int \Phi(\lambda \mathbf{w}^\top \boldsymbol{\phi}_*) P(\mathbf{w}) d\mathbf{w} = \int \Phi(\lambda u) P(u) du \\
&= \int \Phi\left(\frac{v + \boldsymbol{\mu}^\top \boldsymbol{\phi}_* / \sqrt{\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}}{\left(\lambda \sqrt{\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*} \right)^{-1}} \right) P(v) dv \\
&= \Phi\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{\phi}_* / \sqrt{\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}}{\sqrt{1 + \left(\lambda \sqrt{\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*} \right)^{-2}}} \right) \\
&= \Phi\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{\phi}_*}{\sqrt{\lambda^{-2} + \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}} \right) \approx \sigma\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{\phi}_*}{\sqrt{1 + \lambda^2 \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}} \right) \\
&= \sigma\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{\phi}_*}{\sqrt{1 + \frac{\pi}{8} \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}} \right),
\end{aligned}$$

where $u = \mathbf{w}^\top \boldsymbol{\phi}_* \sim \mathcal{N}\left(\boldsymbol{\mu}^\top \boldsymbol{\phi}_*, \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_* \right)$, $v = \frac{u - \boldsymbol{\mu}^\top \boldsymbol{\phi}_*}{\sqrt{\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}} \sim \mathcal{N}(0, 1)$. And the following integral is used

$$\int \Phi\left(\frac{w - a}{b} \right) P(w) dw = \Phi\left(\frac{-a}{\sqrt{1 + b^2}} \right). \quad (12)$$

To prove the above equation, let $X \sim \mathcal{N}(a, b^2)$ and $Y \sim \mathcal{N}(0, 1)$ be independent random variables. Then,

$$P(X \leq Y \mid Y = w) = P(X \leq w) = \Phi\left(\frac{w - a}{b} \right), \quad (13)$$

$$\begin{aligned}
P(X \leq Y) &= \int P(X \leq Y \mid Y = w) P(w) dw \\
&= \int \Phi\left(\frac{w - a}{b} \right) P(w) dw.
\end{aligned} \quad (14)$$

Since $P(X \leq Y) = P(X - Y \leq 0)$ and $X - Y \sim \mathcal{N}(a, b^2 + 1)$, $P(X \leq Y) = \Phi\left(\frac{-a}{\sqrt{1 + b^2}} \right)$. Therefore, (12) holds. □

When it converges, $\det(\boldsymbol{\Sigma}) \approx 0$, we have

$$\mathbb{E}[g_*(\mathbf{w})] \approx \sigma\left(\boldsymbol{\mu}^\top \boldsymbol{\phi}_* \right), \quad (15)$$

where $\boldsymbol{\mu}^\top \boldsymbol{\phi}_*$ is the log-odds of occupancy at $\mathbf{x}_*$.

B. Gradient and Surface Normal Prediction

We can also derive the gradient of (11) as

$$\begin{aligned}
&\nabla \mathbb{E}[g_*(\mathbf{w})] = \sigma(h)\sigma(-h) \nabla_{\mathbf{x}} \boldsymbol{\phi}_* \nabla_{\boldsymbol{\phi}_*} h, \\
&\nabla_{\boldsymbol{\phi}_*} h = \frac{1}{\sqrt{1 + \frac{\pi}{8} \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*}} \left(\boldsymbol{\mu} - \frac{\frac{\pi}{8} \boldsymbol{\mu}^\top \boldsymbol{\phi}_* \boldsymbol{\Sigma} \boldsymbol{\phi}_*}{1 + \frac{\pi}{8} \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_*} \right),
\end{aligned} \quad (16)$$

where $\nabla_{\mathbf{x}} \boldsymbol{\phi}_* = -2\gamma \mathbf{X} \operatorname{diag}(\boldsymbol{\phi}_*)$, $\mathbf{X} = [\mathbf{x}_* - \tilde{\mathbf{x}}_1, \cdots, \mathbf{x}_* - \tilde{\mathbf{x}}_M, \mathbf{0}]$ for the RBF kernel and M hinge points $\{\tilde{\mathbf{x}}_i\}_{i=1}^M$.

When $\det(\boldsymbol{\Sigma}) \approx 0$, the gradient of the occupancy probability at $\mathbf{x}_*$ is approximated as

$$\mathbf{v}_* \approx \sigma(\boldsymbol{\mu}^\top \boldsymbol{\phi}_*)\sigma(-\boldsymbol{\mu}^\top \boldsymbol{\phi}_*)\nabla_{\mathbf{x}} \boldsymbol{\phi}_* \boldsymbol{\mu}. \tag{17}$$

The surface normal at $\mathbf{x}_*$ is the opposite direction of $\mathbf{v}_*$. Since we do not care about the magnitude of $\mathbf{v}_*$ when calculate the surface normal, we have

$$\mathbf{n}_* = -\frac{\mathbf{z}(\boldsymbol{\mu})}{\|\mathbf{z}(\boldsymbol{\mu})\|_2}, \quad \mathbf{z}(\boldsymbol{\mu}) = \nabla_{\mathbf{x}} \boldsymbol{\phi}_* \boldsymbol{\mu}, \tag{18}$$

and the variance is

$$\mathbb{V}[\mathbf{n}_*] = \mathbf{G}^\top \boldsymbol{\Sigma} \mathbf{G}, \quad \mathbf{G} = (\nabla_{\mathbf{x}} \boldsymbol{\phi}_*)^\top \frac{\mathbf{I} - \mathbf{n}_* \mathbf{n}_*^\top}{\|\mathbf{z}(\boldsymbol{\mu})\|_2}. \tag{19}$$

III. SDF Gradient Variance Calculation

Based on the softmin approximation, we can obtain the approximation of the UDF gradient:

$$\begin{aligned} \nabla d(\mathbf{x}_*) &\approx \nabla_{\mathbf{x}_*} h(\mathbf{x}_*, \{\mathbf{x}_i\}_{i=1}^N) = g(\mathbf{x}_*, \{\mathbf{x}_i\}_{i=1}^N) \\ &= \mathbf{V}\left(\alpha \mathbf{s}\mathbf{s}^\top \mathbf{z} - \alpha \operatorname{diag}(\mathbf{s})\mathbf{z} + \mathbf{s}\right), \\ \mathbf{V} &= \left[\mathbf{v}_1 \cdots \mathbf{v}_N\right], \mathbf{v}_i = \frac{\mathbf{x}_* - \mathbf{x}_i}{z_i}. \end{aligned} \tag{20}$$

Therefore, assuming isotropic variance $\mathbb{V}_i$ for each dimension of $\mathbf{x}_i$, we can calculate the variance of $\nabla d(\mathbf{x}_*)$ approximately by Taylor expansion and get (16):

$$\mathbb{V}[\nabla_k u(\mathbf{x}_*)] \approx \sum_{i=1}^N \left\|\nabla_{\mathbf{x}_i} g_k(\mathbf{x}_*, \{\mathbf{x}_i\}_{i=1}^N)\right\|_2^2 \sigma_i^2, 1 \le k \le n,$$

where

$$\begin{aligned} &\nabla_{\mathbf{x}_i} g(\mathbf{x}_*, \{\mathbf{x}_i\}_{i=1}^N) = \\ &\alpha \mathbf{v}_i \left(l_i\left(\mathbf{v}_i - \mathbf{V}\mathbf{s}\right) + s_i(\mathbf{v}_i - \mathbf{g})\right)^\top + \frac{l_i}{z_i}\left(\mathbf{v}_i \mathbf{v}_i^\top - \mathbf{I}\right), \\ &l_i = s_i(\alpha \mathbf{s}^\top \mathbf{z} - \alpha z_i + 1). \end{aligned} \tag{21}$$

IV. Adaptive Scaling for GP Stability

While larger λ (smaller kernel scale) improves the approximation of (8), it triggers numerical underflow of $k(\mathbf{x}_*, \mathbf{x}_i)$. To address this, we introduce a scaling factor γ and rewrite (9):

$$\begin{bmatrix} \hat{f}'(\mathbf{x}_*) \\ \nabla \hat{f}'(\mathbf{x}_*) \end{bmatrix} = \begin{bmatrix} \mathbf{k'}_*^\top \\ \nabla_{\mathbf{x}_*}^\top \mathbf{k'}_* \end{bmatrix} (\mathbf{K} + \boldsymbol{\Sigma}_y)^{-1} \mathbf{1}, \tag{22}$$

where $k'_{*,i} = k'(\mathbf{x}_i, \mathbf{x}_*) = \gamma k(\|\mathbf{x}_i - \mathbf{x}_*\|_2)$. $\gamma > 0$ can be merged into the exponent. By setting γ based on the distance $d = \|\mathbf{x}_* - \mathbf{x}_1\|_2$, we effectively normalize the kernel values and prevent underflow. We set $\gamma = e^{d^2/(2l^2)}$ for the RBF kernel or $\gamma = e^{\sqrt{3}d/l}$ for the Matérn 3/2 kernel. Then, the corresponding inverse transformations $r(\hat{f}')$ is adjusted to $r(\hat{f}') = \sqrt{-2l^2 \log \hat{f}' + d^2}$ for RBF and $r(\hat{f}') = -\frac{l}{\sqrt{3}} \log \hat{f}' + d$ for Matérn 3/2 kernel.

V. Limit Analysis of the Softmin Approximation

This section provides the full derivation summarized by (13) and (14) in Sec. IV-B2, showing that the log-GP posterior mean behaves as a softmin over the surface set in both the weakly- and strongly-correlated regimes of a noisy training set.

Let $\mathbf{K} \in \mathbb{R}^{N \times N}$ with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$. Because every training label in $\mathcal{D}_{\text{surf}}$ is $f(\mathbf{x}_i) = 1$, the posterior mean reduces to $\hat{f}_* = \mathbf{k}_*^\top (\mathbf{K} + \boldsymbol{\Sigma})^{-1} \mathbf{1}$.

A. Weakly-correlated regime

When the kernel scale is small relative to the sample spacing, $K_{ij} \approx 0$ for $i \neq j$, so $\mathbf{K} \to \mathbf{I}$ and

$$(\mathbf{K} + \boldsymbol{\Sigma})^{-1} \mathbf{1} \approx \frac{\mathbf{1}}{1 + \sigma^2}, \tag{23}$$

$$\hat{f}_* \approx \frac{1}{1 + \sigma^2} \sum_i k_{*i} = \frac{S}{1 + \sigma^2} \max_i k_{*i}, \tag{24}$$

where $S = \sum_i k_{*i} / \max_i k_{*i} \ge 1$ is the ratio of the sum of kernel values to the maximum kernel value, which accounts for the inflation effect in the softmin approximation. Taking the logarithm gives (13):

$$\log \hat{f}_* \approx \log \max_i k_{*i} + \log S - \log(1 + \sigma^2), \tag{25}$$

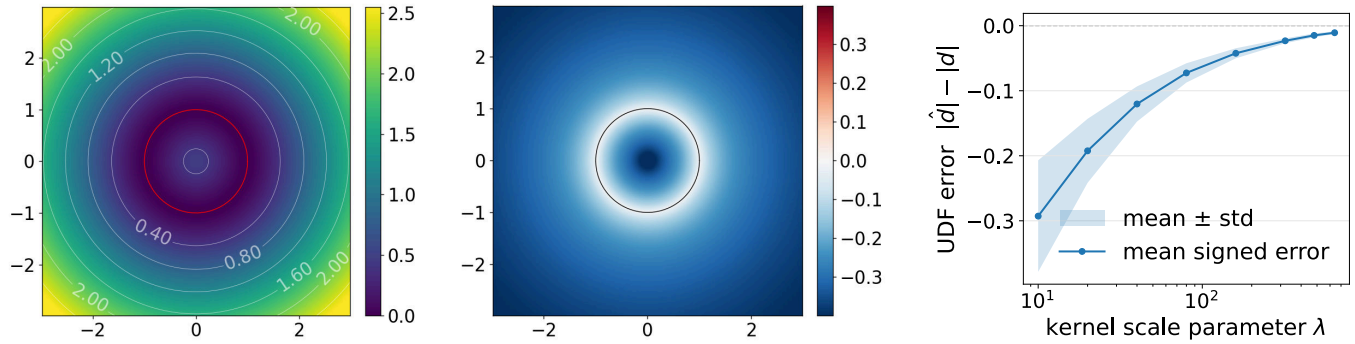
which shows that the log-GP posterior mean is approximately a softmin function, and the UDF is recovered via the kernel-specific nonlinear transform $r(\hat{f})$ in Table I.

Two opposite biases act on the recovered UDF. Because the softmin sums over all nearby surface samples, $\sum_i k_{*i} \ge \max_i k_{*i}$, so $\hat{f}_*$ is slightly inflated and, through the decreasing transform r, the recovered distance is biased *below* the true distance. This is a deterministic recovery bias, present even in the noise-free case, that shrinks as the kernel scale decreases. The noise variance σ^2 acts in the opposite direction: it shrinks $\hat{f}_*$ by $1/(1+\sigma^2)$, biasing the recovered distance slightly *above* the true distance, and additionally down-weights uncertain samples. In practice, the recovery bias dominates, so the net prediction is a conservative under-estimate, which is beneficial for safety.

Taking the RBF kernel as an example, with reverse transform $\hat{d} = r(\hat{f}) = l\sqrt{-2 \log \hat{f}}$,

$$\hat{d} \approx \sqrt{\underbrace{-2l^2 \log \max_i k_{*i}}_{d^2 \text{ (true UDF)}} \underbrace{- 2l^2 \log S}_{\le 0 \text{ (recovery)}} + \underbrace{2l^2 \log(1 + \sigma^2)}_{\ge 0 \text{ (noise)}}}. \tag{26}$$

The recovery term pulls $\hat{d}$ below the true distance d, while the noise term pushes it above. Since $S \ge 1 + \sigma^2$ in practice, the net bias is a conservative under-estimate, and both terms shrink with the kernel scale l. A noise-free recovery study ($\sigma^2 = 0$, Fig. 1) confirms this: the predicted UDF (Fig. 1a) stays below the true distance (Fig. 1b), and the gap shrinks as λ grows (Fig. 1c), matching the recovery term.



(a) (b) (c)

Fig. 1: Noise-free ($\sigma^2 = 0$) benchmark of the log-GP UDF on a 2D unit-circle. (a) Predicted UDF map at $\lambda = 10$; the red curve is the unit circle. (b) Signed error (prediction minus ground truth): negative everywhere off the surface, a pure under-estimate. (c) Error vs. λ: the bias stays negative and shrinks as λ grows, matching the recovery term $-2l^2 \log S$.

B. Strongly-correlated regime

In the dense-sampling case, $K_{ij} \to 1$ so $\mathbf{K} \to \mathbf{1}\mathbf{1}^\top$, and by the Sherman–Morrison formula,

$$(\mathbf{K} + \boldsymbol{\Sigma})^{-1}\mathbf{1} \approx (\sigma^2\mathbf{I} + \mathbf{1}\mathbf{1}^\top)^{-1}\mathbf{1} = \frac{\mathbf{1}}{\sigma^2 + N}, \tag{27}$$

$$\hat{f}_* \approx \frac{1}{\sigma^2 + N}\sum_i k_{*i} = \frac{S}{\sigma^2 + N}\max_i k_{*i}, \tag{28}$$

where $S = \sum_i k_{*i} / \max_i k_{*i}$ as before. Here the tightly-clustered samples have nearly equal kernel values, so $S \to N$ (maximal inflation) and $\hat{f}_*$ is approximately the arithmetic mean of the k_{*i}. This is the $\alpha \to 0$ degeneracy of the softmin (uniform weights): the kernel bumps merge into a single wide bump. For a query at distance $\sim z$ from a tight cluster, all $\|\mathbf{x}_* - \mathbf{x}_i\| \approx z$, so $\hat{f}_* \approx \frac{N}{\sigma^2+N}k(z)$ and, taking the logarithm, we obtain (14):

$$\begin{aligned}\log \hat{f}_* &\approx \log k(z) + \log S - \log(\sigma^2 + N) \\ &\approx \log k(z) - \log\left(\frac{\sigma^2}{N} + 1\right),\end{aligned} \tag{29}$$

where the second step uses $S \to N$. The kernel-specific transform $r(\hat{f})$ still recovers z. The recovery inflation $\log S \to \log N$ is now absorbed by the N-sample normalization $\log(\sigma^2 + N)$, so the deterministic under-estimate cancels and only the noise over-estimate $\log(\frac{\sigma^2}{N} + 1)$ remains. This bias is much smaller than $\log(1 + \sigma^2)$ in the weakly-correlated case, indicating that highly clustered samples are more confident; in practice $N \gg \sigma^2$ and the bias becomes negligible.

C. Intermediate regime

The residual gap lives in the intermediate regime, where $(\mathbf{K} + \boldsymbol{\Sigma})^{-1}\mathbf{1}$ produces weights that are neither uniform nor a nearest-sample indicator. Combining the two limits, the posterior mean can be written approximately as

$$\hat{f}_* \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2 + 1} k(\|\mathbf{x}_* - \mathbf{x}_i\|), \tag{30}$$

for N clusters of different sizes and correlation levels, where σ_i^2 is the equivalent noise variance of the i-th cluster and is smaller for denser or more confident clusters (e.g., $\sigma_i^2 = \sigma^2/N_i$ when the i-th cluster of N_i samples is strongly correlated). The prediction remains a weighted average of kernel bumps dominated by the near samples, with the per-sample noise term σ_i^2 down-weighting uncertain samples, a desirable regularization effect. In the noise-free case $\sigma_i^2 = 0$, the GP posterior mean reduces exactly to the softmin, leaving only the deterministic recovery under-estimate.

VI. Consistency of Min-Fused Local GPs

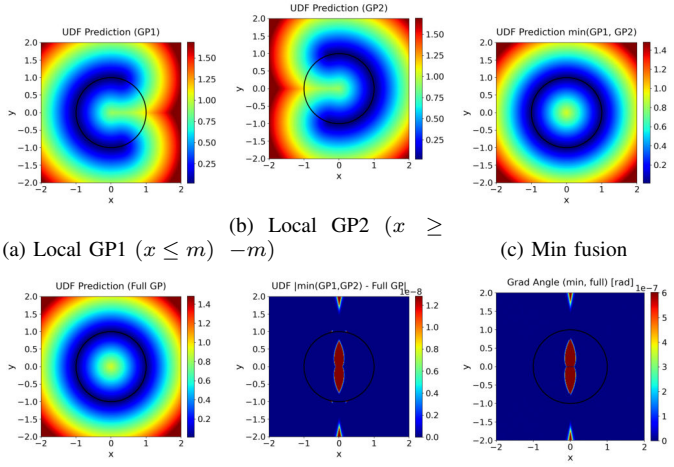
This section provides the 2D demonstration supporting the footnote in Sec. IV-B1, that min-fusing overlapping local GPs matches a single global GP in both value and gradient across octant boundaries.

The training samples are drawn uniformly along a unit circle and partitioned by their x-coordinate into two overlapping subsets: local GP1 is trained on the samples with $x \le m$ and local GP2 on those with $x \ge -m$, so that the two GPs share an overlap band of width $2m$ (here $m = 0.3$). We compare the UDF predicted by each local GP, their minimum fusion, and a single global GP trained on all samples (Fig. 2). The minimum fusion closely matches the global GP: across the entire domain, the UDF difference stays at the 10^{-8} level and the angular error between the predicted gradients remains below 10^{-6} rad, confirming that the overlap preserves both value and gradient consistency, including at the GP-switching boundary.

VII. Priority-Based Update Scheme

In Kernel-SDF, several operations take significant time, including BHM EM update, BHM marching, GP training data collection, and GP training. However, not all operations need to be performed at every time step. Therefore, we introduce a priority-based update scheme to efficiently allocate computational resources to the most needed updates.

To maintain the responsiveness of the system to the latest sensor data, BHM EM updates are performed at every time step for all BHMs that have new sensor data. Newly created BHMs also need to be marched immediately to get surface points for its corresponding GP training data buffer update.



(a) Local GP1 ($x \leq m$) (b) Local GP2 ($x \geq -m$) (c) Min fusion

(d) Global GP (e) $|\text{Min} - \text{Global}|$ (f) Gradient angle diff.

Fig. 2: 2D consistency demo of minimum-fusion local GPs versus a single global GP. Two overlapping local GPs (a), (b) are minimum-fused (c) and compared against a global GP trained on all samples (d). The fusion is almost identical to the global GP: the absolute UDF difference (e) is at the 10^{-8} level and the gradient angular error (f) is below 10^{-6} rad across the whole domain, including the GP-switching boundary. The black curve marks the zero-level set (the unit circle).

For all GP training operations, we delay them until they are required for SDF prediction.

For other operations, we use priority queues to determine the order of updates based on their necessity. In order to make the system also responsive to the latest query trends (e.g., the down-stream planner may focus on a specific area), each GP has a query counter c_0 that records how many times the GP is requested for SDF prediction. The counter helps to reshape the priority of GP training and data buffer updates. When a GP is trained, its query counter is reduced by half. And the query counter is capped to a maximum value (10000) to avoid over-prioritization.

A. Marching Priority Queue

Note that a single EM update of a BHM only affects a small portion of the surface points. Especially for well-updated BHMs, the surface point changes are even smaller. Hence, we first introduce a priority queue for marching BHMs, which orders the BHMs by the latest timestamp $t_{\mathcal{B}}$ when a BHM is requested to update surface estimation. The BHM with the oldest timestamp (smaller $t_{\mathcal{B}}$) in the queue is processed first because it is more likely to be unchanged in the near future:

$$s_{\text{march}} = t_{\mathcal{B}} \downarrow, \tag{31}$$

where $\downarrow$ indicates smaller values have higher priority.

B. GP Data Buffer Update Queue

After marching a BHM, its corresponding GP training data buffer needs to be updated with the new surface points. However, it also takes time to get significant changes of $\mathcal{D}_{\text{surf}}$ for updating the GP. And $\mathcal{D}_{\text{surf}}$ is not required until the GP needs to be trained. Therefore, we introduce a priority queue that orders the GPs by the following score:

$$s_{\text{buffer update}} = c_1(1 + \eta_1 c_0) \uparrow, \tag{32}$$

where c_1 is the number of times the GP data buffer is marked to be updated since the last update, η_1 is a weight parameter, and $\uparrow$ indicates larger values have higher priority. c_1 is increased by 1 each time the GP is marked to update its data buffer after marching its corresponding BHM and reset to 0 after the data buffer is updated. For newly created GPs, we initialize c_1 to $c_1^{\max} \exp(-\gamma \|\mathbf{c}_{GP} - \mathbf{o}_t\|_2)$, where $\mathbf{c}_{GP}$ is the center of the GP training data collection bounding box, $\mathbf{o}_t$ is the current sensor origin, and $\gamma > 0$ is a decay parameter. This initialization gives higher priority to GPs closer to the sensor origin.

C. GP Training Queue

Similar to the GP data buffer update, GP training is not required until SDF prediction is requested. In addition, training a GP only makes sense when its training data buffer has significant changes. Therefore, we introduce another priority queue that orders the GPs by the following score:

$$s_{\text{GP train}} = c_2(1 + \eta_2 c_0) \uparrow, \tag{33}$$

where c_2 is the number of times the GP is marked to be trained since the last training, η_2 is a weight parameter, and $\uparrow$ indicates larger values have higher priority. c_2 is increased by c_1 each time the GP's buffer is updated and reset to 0 after the GP is trained. When the GP is trained, c_0 is also halved in case the GP is over-prioritized for future updates as the query trend may change.

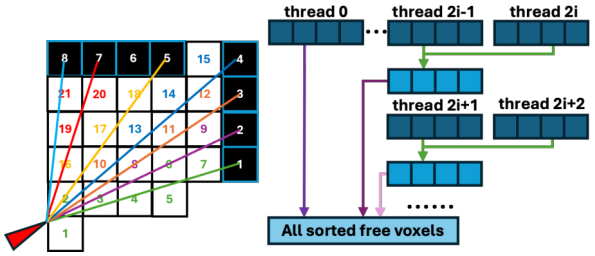
VIII. Efficient Tree Construction

To optimize occupancy mapping, we extend the Octomap [3] to include a dedicated 2D quadtree implementation. Performance is further boosted by updating voxels in a sorted, along-ray order (Fig. 3a). This approach leverages spatial locality that when a free voxel is updated, its neighboring voxels are more likely to be updated, improving the CPU cache hit rate. While the order of occupied voxels is obvious, we implement an algorithm demonstrated in Fig. 3b to efficiently generate the sorted free-voxel sequence.

IX. Experiment Details

A. Parameter Settings

For Kernel-SDF, we set the octree resolution to 0.08m, 0.05m and 0.7m for the Replica, Cow&Lady, and Newer College datasets, respectively. The local BHMs are placed at the depth $D_{\text{tree}} - 1$, which means the local BHM bounding box size is twice the octree voxel size. The number of hinge points per axis is set to 7 for the Replica and Cow&Lady datasets, and 9 for the Newer College dataset. The BHM kernel scale l is set to 0.016m, 0.013m and 0.11m for the three datasets, respectively. For the GP, we use RBF kernel and set $\lambda = \frac{1}{2l^2}$ to 500, 300, and 250 for the three datasets, respectively. For other parameters, please refer to our released code.



(a) Voxel update order (b) Free voxel parallel generation

Fig. 3: To construct the tree efficiently, we implement two key optimizations: (a) voxels are updated in ray-sorted order to maximize the CPU cache hit rate; and (b) this ordering is generated via a stride-2 parallel reduction. In this scheme, threads first process ray batches in parallel, then odd-indexed threads $(2i-1)$ merge results from their even-indexed counterparts $(2i)$. Finally, these partially merged results are aggregated sequentially into thread 0.

For the baselines, we use the default parameters provided in their released code except for necessary modifications to adapt to the datasets. For example, iSDF [4] only supports image input, so we convert the LiDAR scans of the Newer College dataset to depth images using the camera intrinsic parameters that give the same field of view as the LiDAR. We set the voxel size of Voxblox and FIESTA to be 0.05m for the Replica and Cow&Lady datasets, and 0.2m for the Newer College dataset to get a better trade-off between accuracy and efficiency.

For our method and VDB-GPDF [5], a mesh is available from the algorithm directly. For Voxblox [6], FIESTA [7], and iSDF [4], we extract the mesh using the marching cubes algorithm [8] with the voxel size as the marching cubes resolution, except for iSDF, where we use a resolution of 0.02m for the Replica and Cow&Lady datasets and 0.1m for the Newer College dataset.

B. Mesh Metrics

For Replica dataset, we sample 2 million points from the ground truth mesh and the reconstructed mesh, respectively. For Cow&Lady dataset, we sample 54k points from the ground truth point cloud and the reconstructed mesh, respectively. For Newer College dataset, we sample 2 million points from the ground truth point cloud and the reconstructed mesh, respectively.

We compute precision, recall, and F1 score with a threshold of 0.05m for the Replica and Cow&Lady datasets, and 0.2m for the Newer College dataset. A point is considered to be correctly reconstructed if its distance to the other point cloud is less than the threshold.

Besides, we compute the accuracy and the completion, which are defined as:

$$\text{Accuracy} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{q}\in\mathcal{Q}} \|\mathbf{p}-\mathbf{q}\|_2, \tag{34}$$

$$\text{Completion} = \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{q}\in\mathcal{Q}} \min_{\mathbf{p}\in\mathcal{P}} \|\mathbf{p}-\mathbf{q}\|_2, \tag{35}$$

where $\mathcal{P}$ is the reconstructed point cloud and $\mathcal{Q}$ is the ground truth point cloud.

We also choose to use Chamfer-L1 distance in place of the standard Chamfer distance for more intuitive interpretation of the error in meters, which is the average of accuracy and completion.

C. SDF Metrics

For SDF evaluation, we query each method with a regular grid of points covering the bounding box of the ground truth mesh or point cloud. The grid resolution is set to 0.05m for the Replica and Cow&Lady datasets, and 0.2m for the Newer College dataset. Prediction may fail for Voxblox and FIESTA at some query points because they rely on the eight voxel corners to interpolate the SDF value. In such cases, we exclude the failed points when calculating the metrics for Voxblox and FIESTA.

D. Time Metrics

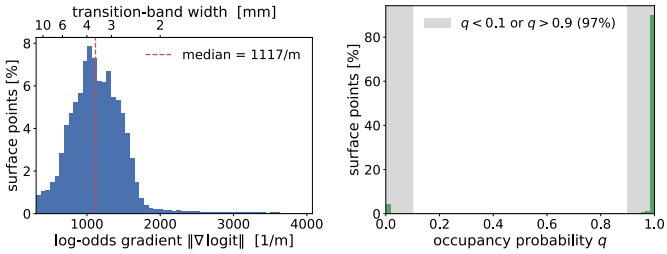
While it is straightforward to measure the prediction time for all methods, measuring the update time is more complex due to the different update strategies employed by each method. For a fair comparison, we measure the average update time per scan for all methods. Specifically, we record the total time taken to process all scans in a dataset (but exclude time of data loading, logging and etc.) and divide it by the number of scans to obtain the average update time per scan. This approach provides a consistent basis for comparing the efficiency of each method's update process.

X. Full Sign-Prediction Metrics

This section provides the full quantitative support for the deterministic-sign treatment discussed in Sec. IV-B3 and the sign-accuracy summary in Sec. V-B, including the empirical sharp-transition evidence, the sign-mixture variance derivation, and the complete per-scene sign-metric table.

A. Sharp-Transition Evidence

BHM provides a probabilistic sign prediction $P(s(\mathbf{x}) = 1) = q$ and $P(s(\mathbf{x}) = -1) = 1 - q$, where $s(\mathbf{x})$ is the sign at query $\mathbf{x}$. To substantiate the sharp-transition assumption empirically, we evaluate the learned BHM occupancy field at all ground-truth surface vertices of the Replica `room0` scene (Fig. 4). At the surface, the occupancy log-odds gradient $\nabla \log \frac{q}{1-q}$ has a median norm of $\approx 1.1 \times 10^3\,\mathrm{m}^{-1}$ (Fig. 4a). Equivalently, the occupancy probability rises from 0.12 to 0.88 over a transition band of median width $\approx 3.6\,\mathrm{mm}$ (95th percentile $\approx 7.4\,\mathrm{mm}$), far below the map resolution. As a direct consequence, 97% of the surface points already have $q < 0.1$ or $q > 0.9$ (Fig. 4b): the occupancy probability saturates within a sub-voxel shell. So, for any query slightly off the surface, q is effectively 0 or 1, which justifies the deterministic-sign treatment.



(a) (b)

Fig. 4: Sharp transition of the BHM occupancy field at the ground-truth surface vertices of the Replica `room0` scene. (a) Occupancy log-odds gradient norm $\|\nabla \log \frac{q}{1-q}\|$, with the equivalent transition-band width (top axis, median ≈ 3.6 mm). (b) Occupancy probability q at the surface: 97% of points have $q < 0.1$ or $q > 0.9$ (shaded), so the sign is effectively deterministic.

B. Sign-Mixture Variance

Under the probabilistic sign, the SDF prediction $\hat{d}(\mathbf{x})$ has a mixture distribution with density

$$p(\hat{d}(\mathbf{x})) = q\, p(\hat{u}(\mathbf{x}) \mid s(\mathbf{x}) = 1) + (1-q)\, p(\hat{u}(\mathbf{x}) \mid s(\mathbf{x}) = -1), \tag{36}$$

where $\hat{u}(\mathbf{x})$ is the predicted unsigned distance at $\mathbf{x}$. Because BHM learns a continuous occupancy field with a sharp transition at the surface, q is close to either 0 or 1, so the mixture collapses to $p(\hat{d}(\mathbf{x})) = p(\hat{u}(\mathbf{x}))$. Thus $\mathbb{E}[\hat{d}(\mathbf{x})] = s(\mathbf{x})\mathbb{E}[\hat{u}(\mathbf{x})]$ and $\mathbb{V}[\hat{d}(\mathbf{x})] = \mathbb{V}[\hat{u}(\mathbf{x})]$: the uncertainty of the SDF prediction is solely determined by the UDF prediction from the GP back-end, and the deterministic-sign treatment introduces no additional uncertainty in practice. For the $\sim 3\%$ of near-surface points without a sharp transition ($q \in (0.1, 0.9)$), the exact mixture variance contains an additional term proportional to $q(1-q)\mathbb{E}[\hat{u}]^2$ that this treatment discards, so the predictive variance is theoretically underestimated in that near-surface subset.

C. Full Per-Scene Sign Metrics

Table I reports a complete evaluation of the sign prediction on the Replica dataset, where each method's predicted sign is compared against the ground-truth sign extracted from the ground-truth mesh. Ground-truth sign for Cow & Lady and Newer College is unavailable due to the lack of ground-truth meshes. We report F1 score, precision, recall, and accuracy separately for points near the surface (Near), far from the surface (Far), and all points (All). Far from the surface the sign is unambiguous and every method exceeds 99% on all metrics, so the near-surface region is the discriminating one, since this is where sign errors are both most likely and most consequential for downstream planning.

Treating *free* as the positive class, precision is a safety-critical quantity because it is the fraction of points declared free that are truly free; high precision corresponds to a low false-free (collision) rate. In the near region, our BHM front-end attains the highest precision, ranking first in 5/8 scenes and second in the remaining 3, and stays above 99% in every scene. Its recall and accuracy are not the best: FIESTA and iSDF rank higher because they label more points as free, but our scores remain consistent across all eight scenes and trade some recall for a substantially lower false-free rate, which is desirable for a conservative overestimation of occupied space. VDB-GPDF makes the opposite trade-off: it reaches perfect recall (100%) by predicting nearly all near-surface points as free, but its near-surface precision drops to 69–82%, giving it the highest false-free rate among all methods. This confirms that recall alone is misleading and that precision (false-free rate) must be reported to expose sign mistakes.

References

[1] T. S. Jaakkola and M. I. Jordan, "A variational approach to bayesian logistic regression models and their extensions," in *AISTATS*, 1997.

[2] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *CoRL*, 2017.

[3] A. Hornung, *et al.*, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[4] J. Ortiz, *et al.*, "iSDF: Real-time neural signed distance fields for robot perception," in *RSS*, 2022.

[5] L. Wu, *et al.*, "VDB-GPDF: Online gaussian process distance field with vdb structure," *RA-L*, 2025.

[6] H. Oleynikova, *et al.*, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," in *IROS*, 2017.

[7] L. Han, *et al.*, "FIESTA: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *IROS*, 2019.

[8] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *ACM SIGGRAPH*, 1987.

TABLE I: Sign metrics. Results are reported as the F1 score, precision, recall and accuracy for all points (All), points near the surface (Near) and points far from the surface (Far) on the Replica dataset. The best, second best and third best results are bold, underlined and curly-underlined, respectively.

Metric	Region	Method	room 0	room 1	room 2	office 0	office 1	office 2	office 3	office 4
F1 Score [%] ↑	All	Ours	**98.25**	97.69	97.66	97.00	96.52	97.92	97.79	97.91
		FIESTA	97.74	97.69	97.66	96.74	96.53	98.21	**97.97**	97.83
		Voxblox	96.77	97.91	96.82	97.24	97.35	95.33	94.97	95.46
		iSDF	97.93	**98.43**	**98.26**	**98.60**	**98.09**	**98.34**	97.63	**98.14**
		VDB-GPDF	86.85	86.07	91.54	85.26	86.34	87.90	84.04	87.94
	Near	Ours	**94.51**	93.43	93.26	91.71	91.33	93.83	93.49	93.26
		FIESTA	93.02	93.50	93.20	91.05	91.54	94.65	**93.95**	93.06
		Voxblox	90.72	94.22	91.38	93.75	94.31	88.96	86.34	87.29
		iSDF	93.76	**95.76**	**95.24**	**96.29**	**95.47**	**95.34**	93.29	**94.41**
		VDB-GPDF	84.75	84.71	90.25	83.80	85.52	85.86	81.75	85.87
	Far	Ours	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.98	99.95	**100.00**
		FIESTA	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		Voxblox	99.58	99.93	99.75	99.29	99.49	98.54	99.18	99.22
		iSDF	99.99	99.99	99.99	**100.00**	99.99	99.97	99.93	99.96
		VDB-GPDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Precision [%] ↑	All	Ours	99.81	99.81	**99.90**	99.68	**99.64**	**99.86**	**99.85**	**99.92**
		FIESTA	97.85	98.72	99.15	98.45	97.27	98.85	98.28	98.58
		Voxblox	**99.90**	**99.84**	99.74	**99.74**	99.61	99.80	99.84	99.87
		iSDF	98.73	98.24	98.80	99.09	98.42	98.72	98.90	99.75
		VDB-GPDF	76.76	75.55	84.39	74.31	75.96	78.41	72.47	78.48
	Near	Ours	99.37	99.44	**99.69**	99.06	**99.05**	**99.56**	**99.55**	**99.74**
		FIESTA	93.35	96.32	97.47	95.62	93.26	96.54	94.83	95.39
		Voxblox	**99.67**	**99.54**	99.22	**99.28**	99.03	99.38	99.47	99.53
		iSDF	96.08	95.23	96.66	97.57	96.22	96.34	96.76	99.22
		VDB-GPDF	73.54	73.47	82.24	72.11	74.70	75.22	69.14	75.24
	Far	Ours	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		FIESTA	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		Voxblox	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		iSDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		VDB-GPDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Recall [%] ↑	All	Ours	96.74	95.66	95.52	94.46	93.58	96.06	95.81	95.97
		FIESTA	97.63	96.69	96.20	95.08	95.81	97.57	97.67	97.09
		Voxblox	93.83	96.05	94.07	94.86	95.20	91.24	90.56	91.43
		iSDF	97.15	98.63	97.73	98.12	97.77	97.96	96.39	96.58
		VDB-GPDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	Near	Ours	90.10	88.10	87.61	85.38	84.73	88.73	88.12	87.56
		FIESTA	92.70	90.85	89.28	86.89	89.89	92.84	93.09	90.85
		Voxblox	83.24	89.45	84.69	88.80	90.02	80.52	76.27	77.72
		iSDF	91.54	96.31	93.86	95.05	94.74	94.36	90.06	90.05
		VDB-GPDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	Far	Ours	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.97	99.89	**100.00**
		FIESTA	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		Voxblox	99.17	99.87	99.51	98.58	98.98	97.13	98.38	98.46
		iSDF	99.98	99.98	99.98	**100.00**	99.99	99.93	99.86	99.92
		VDB-GPDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Accuracy [%] ↑	All	Ours	**97.07**	96.36	95.82	95.17	95.27	96.47	96.70	96.42
		FIESTA	96.23	96.37	95.86	94.77	95.26	96.99	**96.98**	96.31
		Voxblox	94.62	96.68	94.31	95.55	96.37	92.16	92.61	92.24
		iSDF	96.47	**97.47**	**96.82**	**97.70**	**97.34**	**97.09**	96.39	**96.73**
		VDB-GPDF	90.55	89.90	93.30	88.29	90.38	90.79	90.43	90.98
	Near	Ours	**92.13**	90.42	89.45	88.16	87.57	91.01	91.05	90.32
		FIESTA	88.24	89.17	88.61	85.27	85.62	90.81	89.68	88.61
		Voxblox	87.52	92.00	86.75	91.40	91.95	84.85	83.34	82.78
		iSDF	91.06	**93.78**	**92.22**	**94.69**	**93.34**	**93.01**	**91.06**	**91.89**
		VDB-GPDF	73.54	73.47	82.24	72.11	74.70	75.22	69.14	75.24
	Far	Ours	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.97	99.89	**100.00**
		FIESTA	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		Voxblox	99.17	99.87	99.51	98.58	98.98	97.13	98.38	98.46
		iSDF	99.98	99.98	99.98	**100.00**	99.99	99.93	99.86	99.92
		VDB-GPDF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**