Supplementary Material for the Paper

Yueji Ma [1,2], Jialu Shen[1], Yanzun Meng[1,2], Dong Xiao[3,4], Zuoqiang Shi[2,5] *, Bin Wang[3] †
[1] Department of Mathematical Sciences, Tsinghua University
[2] Yau Mathematical Sciences Center, Tsinghua University
[3] School of Software, Tsinghua University
[4] School of Mathematical Sciences, University of Science and Technology of China
[5] Yanqi Lake Beijing Institute of Mathematical Sciences and Applications

1 Visualization of Our Method's Reconstruction and Orientation

Inspired by the practical necessity of calculating potentials in diverse electromagnetic media, we add a scaling matrix to the Laplace equation to make it anisotropic and pioneer its applications in surface reconstruction. It ultimately improves the performance of orientation and reconstruction. Due to limitations in the length of the paper, we add more qualitative outputs and comparisons between our method and baselines on orientation and reconstruction in the supplementary materials to show our method's state-of-the-art performance.

1.1 Sparse Point Clouds in Famous Datasets

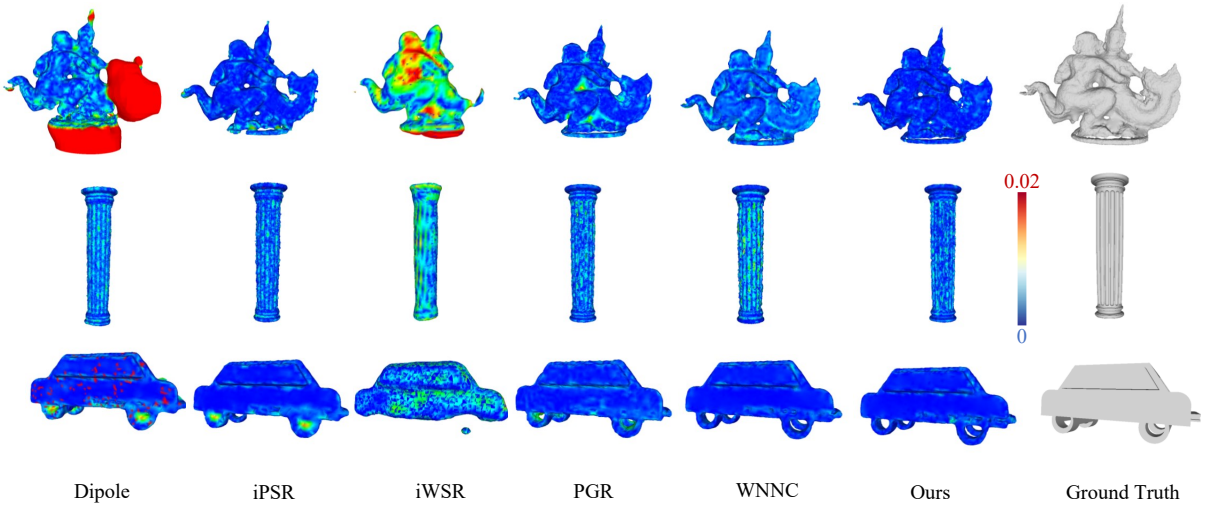

Fig 1: Qualitative comparison of reconstruction for our method and baselines on point clouds in ABC, Real-world, Thingi10k Datasets. The colors, accompanied by a color bar, indicate the Hausdorff distance between the reconstructed surface and the ground truth. Our method outperforms the baselines against reconstruction for sparse point clouds (5K).

ABC, Famous, Thingi10k, and Real-world are famous datasets for reconstruction. These datasets include many different types of point clouds, e.g. daily necessities, fine workpieces, and sculptures, which can measure the model's ability effectively. Our method can accomplish the complex reconstructions for sparse point clouds (5K) that other well-known methods cannot handle well, as shown in Figure 1. These examples include real-world point clouds (the first example of Figure 1) and point clouds with complex details (the second and third examples of Figure 1).

*Co-corresponding author, author's address: zqshi@tsinghua.edu.cn
†Co-corresponding author, author's address: wangbins@tsinghua.edu.cn

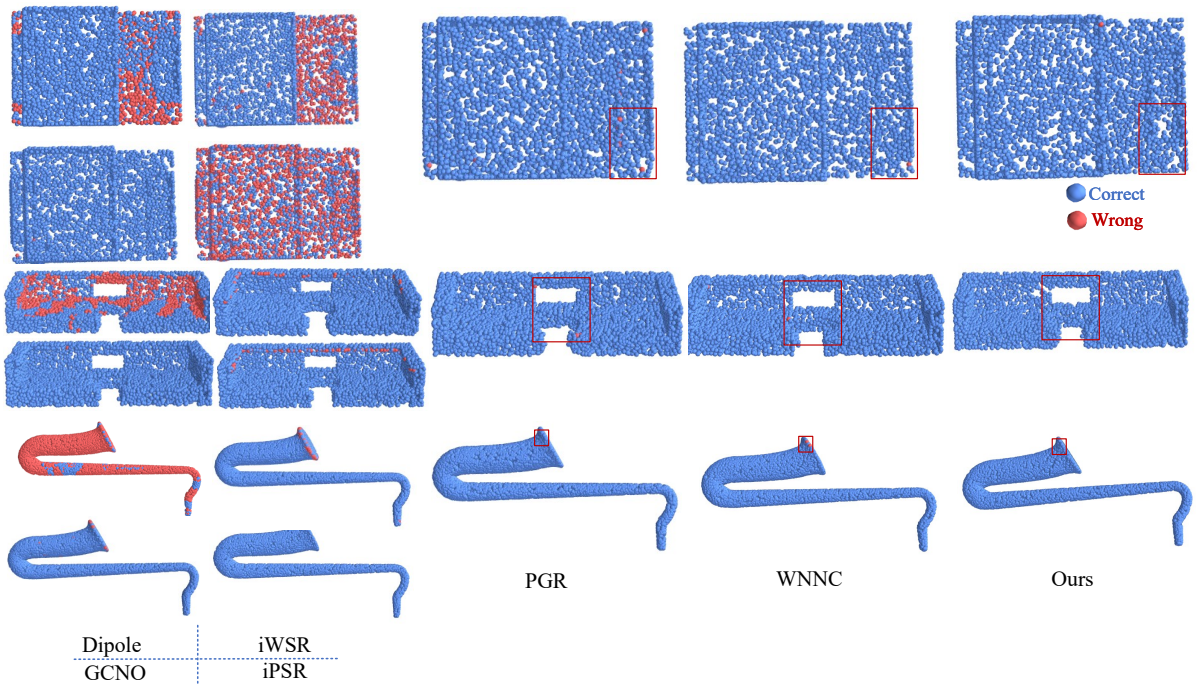

Fig 2: Qualitative comparison of orientation for our method and other state-of-the-art method on the point clouds with small holes or thin structures. Our method outperforms the baselines against orientation.

Due to the introduction of anisotropy, we can construct more non-homogeneous equations to form a linear system with a unique solution. The orientation of sparse point clouds is a very challenging problem. Our method can achieve high-quality orientation of complex point clouds, as shown in Figure 1, including deep cup (the first example of Figure 1), letter text (the second example of Figure 1), and fine workpieces (the third and fourth examples of Figure 1).

We further compare the normal estimation ability of our method and baselines. In Figure 2, the quality of different methods on orientation is shown, with colors representing whether correct or not. Our method demonstrates the state-of-the-art ability. Based on geometric features, iPSR tends to make mistakes in point clouds with thin structures. Our method can handle these issues better based on the PCG algorithm and normal updating operator with the octree-based acceleration.

1.2 Point Clouds with Thin Structures or Small Holes

Point clouds with thin structures or small holes are the most challenging and concerning tasks. PGR heavily relies on regularization terms due to the lack of equations, ultimately leading to adhesion and sealing, as shown in Figure 3. The models in Figure 3 are sampled with 50K points.

Figure 3 shows the comparison of our method with other well-known methods for reconstructing point clouds with thin structures. Due to the influence of regularization terms, PGR suffers from sealing holes. Due to the lack of convergence support for PGR, there may be situations where the model details converge to incorrect solutions. Our method no longer requires any regularization terms. We use the normal updating operator with the octree-based acceleration to further improve normal quality, which can still maintain convergence in theory. Our method can avoid holes and preserve details such as letters (the first example of Figure 3), and protect small holes (the second and third examples of Figure 3).

To demonstrate the improvement of our method's effectiveness by introducing anisotropy. We provide more examples of the reconstruction in Figure 4 to show our method's ability to reconstruct the models with thin structures.

Figure 4 demonstrate that our method consistently upholds high reconstruction quality across various objects such as workpieces, fighter jets, and daily necessities, emphasizing its ability to preserve complex surface details.

Fig 3: Qualitative comparison of reconstruction for point clouds in models with thin structures or small holes. Our method can handle both types of point clouds well and output the high-quality reconstructions.

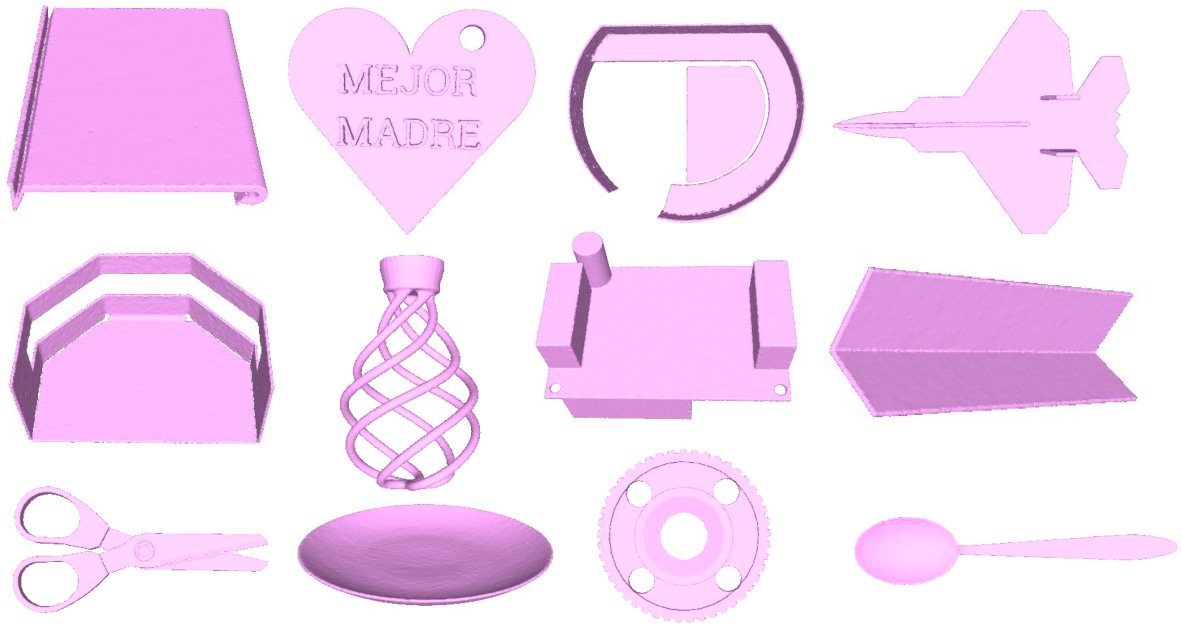

Fig 4: Qualitative result of our method's reconstructions for point clouds with complex thin structures. Our method addresses the limitations of existing methods that are prone to surface damage, enhancing the quality of thin structure reconstructions.

1.3 Ultra Large-scale Point Clouds

Processing ultra-large-scale point cloud datasets demands large amounts of memory and long running time, posing challenges that exceed the capabilities of most state-of-the-art methods. GCNO, iPSR, and PGR exhibit limitations in handling models with more than 1M points. To demonstrate the processing capability of our method for large-scale point clouds, we select a 5 million(5M) point cloud with thin structures in Figure 5.

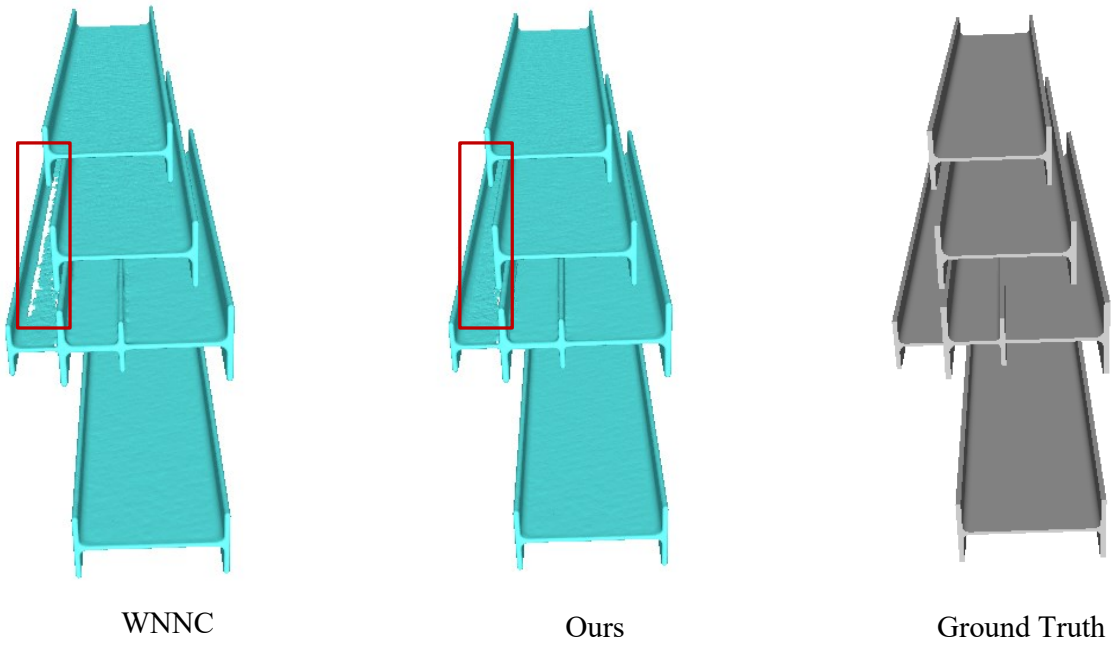

Fig 5: Qualitative comparison of reconstruction for large-scale point clouds. Our method gives the best and more complete reconstruction.

The analysis of Figure 5 reveals that our method handles the models with more than 5M points effectively, exhibiting superior reconstruction quality compared to WNNC. The absence of equations in WNNC restricts the comprehensive utilization of information at thin structural points, leading to some degree of surface degradation.

1.4 Point Clouds from Real Scaning

Real data is a major application prospect and goal for point cloud reconstruction. It tends to include noise and outliers during the scanning process in reality. This poses a higher challenge to the robustness of the method. To showcase the extensive practical applicability of our approach, we have included additional examples of reconstructions in Figure 6 to illustrate our method's proficiency in reconstructing models from real scanning.

Figure 6 shows that our method can effectively reduce the impact of slight noise, eliminate the interference of outliers, and output smooth, accurate surfaces. Moreover, our method also effectively reproduces the details of real objects, such as the gaps between the left and right mouse buttons and remote control buttons.

Based on geometric features, iPSR can only orient at sharp corners instead of estimating the normals well. It tends to make mistakes in point clouds with thin structures. PGR and our method can handle these issues better based on solving equations. Moreover, our method can further improve the details of estimated normals due to the introduction of anisotropy.

1.5 Point Clouds with Noise

Reconstructing noisy point clouds can demonstrate the model's robustness. As our method and PGR rely on regularization terms to solve equations, we further explore our method's reconstruction of point clouds with noise and provide more qualitative examples. m

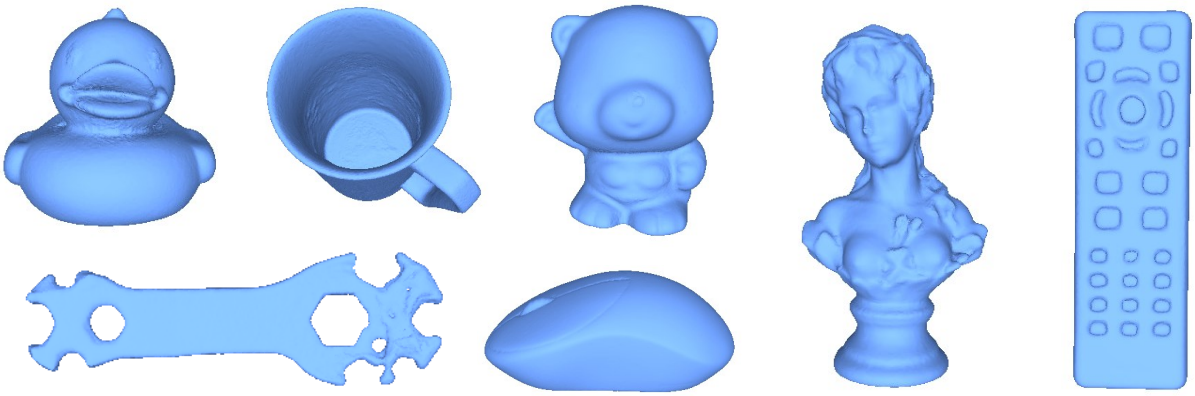

Fig 6: Qualitative result of our method on the reconstruction of point clouds from real scaning. Our method gives smooth and high-quality surfaces.

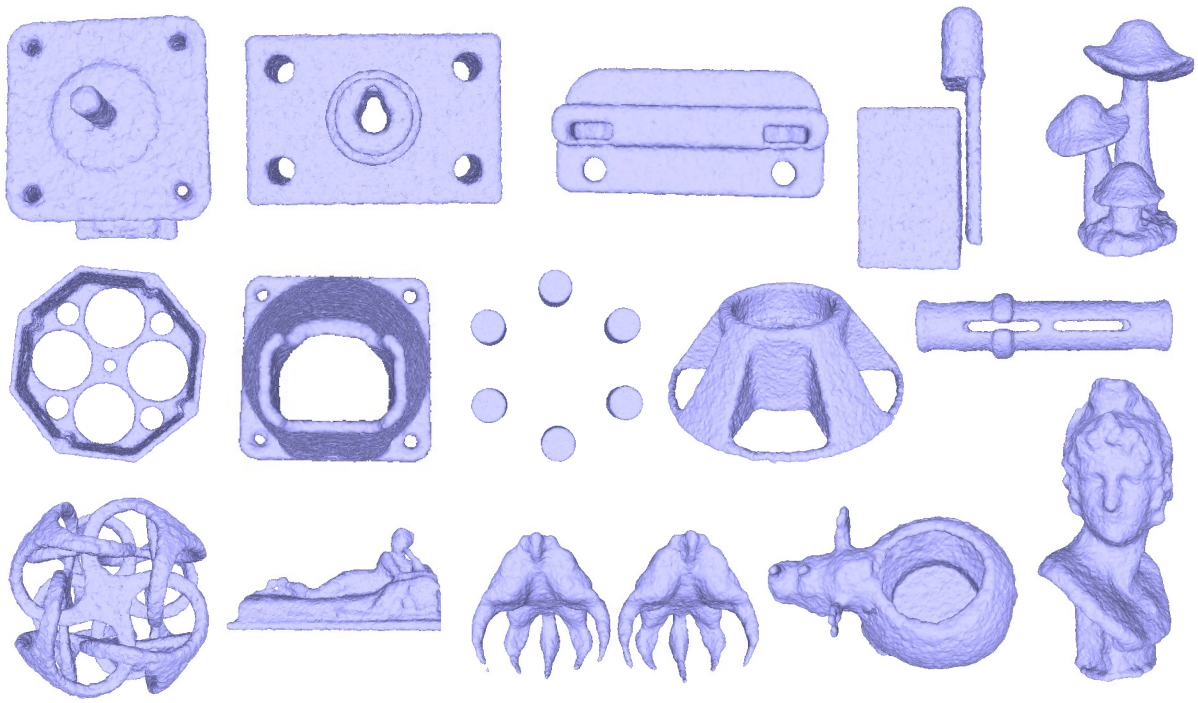

Fig 7: Qualitative result of our method on the reconstruction for point clouds with 0.5% Guassian noise. Our method gives smooth and high-quality surfaces.

Table 1: Quantitative comparisons of WNNC with Anisotropic-WNNC (An-WNNC) on the orientation and reconstruction in both clean and noisy datasets. The Chamfer Distance (CD) values are multiplied by 10^5. The best results are in bold. After introducing anisotropy, the equation contains more information, resulting in an improvement in the quality of orientation and reconstruction compared to the original isotropic WNNC and Gauss Reconstruction.

		5K,noise:0%				5K,noise:0.5%				50K,noise:0.0%		
		Realworld	Famous	ABC	Thingi10k	Realworld	Famous	ABC	Thingi10k	Realworld	Famous	ABC
PGP_{90} ↑	WNNC	0.9976	0.9827	0.9571	0.9803	0.9757	**0.9556**	0.9364	0.9713	**0.9999**	0.9961	0.9922
	An-WNNC	**0.9990**	**0.9830**	**0.9575**	**0.9806**	**0.9843**	0.9553	**0.9394**	**0.9718**	**0.9999**	**0.9969**	**0.9924**
NC_p ↑	WNNC	0.9486	0.9019	0.8677	0.9316	0.8583	0.8005	0.7918	0.8627	0.9841	0.9712	0.9715
	An-WNNC	**0.9528**	**0.9024**	**0.8732**	**0.9323**	**0.8609**	**0.8077**	**0.7953**	**0.8624**	**0.9855**	**0.9753**	**0.9725**
NC_s ↑	WNNC	0.9210	0.9085	0.8656	0.9249	0.8922	0.8830	0.8343	0.8953	0.9539	0.9511	0.9328
	An-WNNC	**0.9217**	**0.9087**	**0.8725**	**0.9249**	**0.8942**	**0.8832**	**0.8366**	**0.8957**	**0.9541**	**0.9521**	**0.9338**
CD ↓	WNNC	25.11	8.12	15.42	11.64	26.32	9.62	17.14	13.12	23.44	7.41	12.42
	An-WNNC	**25.10**	**8.09**	**14.66**	**11.21**	**26.11**	**9.54**	**16.88**	**13.01**	**23.42**	**7.26**	**12.27**

We conduct a large number of experiments, as shown in Figure 7. Owing to the presence of a regularization term, PGR frequently demonstrates a tendency to inaccurately seal fine holes, a phenomenon that becomes pronounced in noisy point clouds. Even when these holes are relatively large, PGR seals them erroneously. Due to introducing anisotropy, our method constructs more non-homogeneous and linearly independent equations. Our method utilizes gradient updates instead of relying on artificial regularization terms, and the improvement is universal and significant.

1.6 Point Clouds with Sharp Edges

Given the challenges posed by reconstructing models with sharp edges, attributed to the limited point cloud density and abrupt variations in indicator functions, the risk of edge blunting or disappearance is heightened. Consequently, we conduct targeted experiments on our proposed method, as depicted in Figure 8. As shown in Figure 8, our method can effectively restore sharp claws and the sharp edge of axes.

1.7 WNNC with Anisotropy

The derived anisotropic Gauss formula can also be used to improve WNNC. Specifically, we can replace A in WNNC with $\left(A_{\boldsymbol{d}_1} + A_{\boldsymbol{d}_2} + A_{\boldsymbol{d}_3}\right)/3$ and maintain its original iteration strategy.

We conducted a comprehensive experimental comparison on the the Famous, ABC, Thingi10K, and Real-world datasets. Table 1 compares sparse point clouds, noisy point clouds, and dense point clouds. It can be seen that the method (anisotropic winding number normal consistency, An-WNNC) can also outperform the original WNNC on both orientation and reconstruction. In the future, we will continue to derive anisotropic normal update iterative operators $\boldsymbol{\mu} \mapsto A_{\boldsymbol{d}_i}\boldsymbol{\mu}$ to achieve the complete anisotropic WNNC (AWNNC) algorithm and further improve the output.

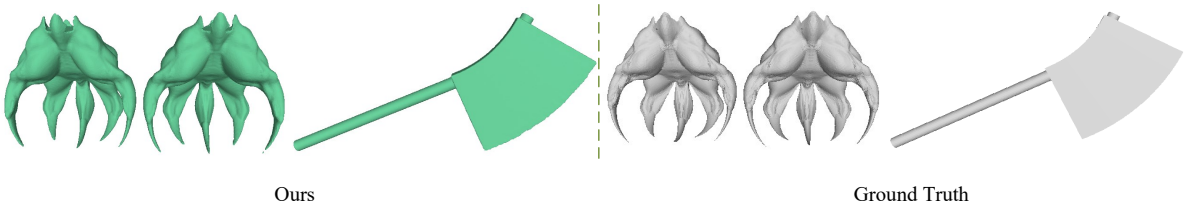

Fig 8: Qualitative result of our method on the reconstruction for point clouds with sharp edges.

2 Experimental Validation of Solvability

We construct a $3N \times 3N$ square matrix by selecting different scaling vectors in the paper. The properties and rank of the corresponding matrix are crucial concerns for us: if it is not full rank, the equations still do not have a unique solution. From the perspective of formulas, taking $\boldsymbol{d}_1 = (3,1,1)^T$,$\boldsymbol{d}_2 = (1,3,1)^T$ as example,

$$\frac{\varphi^{\boldsymbol{d}_1}}{\varphi^{\boldsymbol{d}_2}} = \frac{((y_1-x_1)^2 + \frac{(y_2-x_2)^2}{3} + (y_3-x_3)^3)^{\frac{3}{2}}}{(\frac{(y_1-x_1)^2}{3} + (y_2-x_2)^2 + (y_3-x_3)^3)^{\frac{3}{2}}}.$$

is non-linear and non-proportional for any $\boldsymbol{y}$ and $\boldsymbol{x}$.

From the perspective of numerical experiments, we conducted more extensive experiments on the guarantee for the solvability (full rank) of the linear system. In detail, we selected 5, 10, 50, 100, 500, 1000, and 5000 points from Bunny and 3D spheres respectively(10 experiments for each type), and constructed the linear equation systems according to the strategy in the article(A in Equation 13 of our paper). The Python-based analysis confirmed that all linear systems(140 matrices) were full rank. Taking 5 points on a three-dimensional unit sphere as an example, the constructed matrix is:

$$A = \begin{bmatrix}
0 & 0.08 & 0.03 & 0.21 & 0.29 & 0 & -0.12 & 0.02 & -0.23 & -0.08 & 0 & -0.001 & -0.16 & -0.04 & 0.01 \\
-0.08 & 0 & -0.03 & -0.003 & 0.03 & -0.12 & 0 & -0.06 & -0.06 & -0.07 & 0.001 & 0 & -0.06 & -0.003 & 0.003 \\
-0.03 & 0.03 & 0 & 0.04 & 0.08 & -0.02 & 0.05 & 0 & -0.07 & -0.03 & 0.16 & 0.06 & 0 & 0.08 & 0.07 \\
-0.24 & 0.02 & -0.05 & 0 & 0.22 & 0.27 & 0.28 & 0.10 & 0 & 0.07 & 0.04 & 0.02 & -0.11 & 0 & 0.04 \\
-0.08 & -0.09 & -0.05 & -0.22 & 0 & 0.02 & 0.18 & 0.02 & -0.01 & 0 & -0.003 & -0.001 & -0.04 & -0.04 & 0 \\
0 & 0.14 & 0.03 & 0.24 & 0.07 & 0 & 0.20 & 0.02 & -0.27 & -0.02 & 0 & 0.002 & -0.16 & 0.04 & 0.002 \\
-0.14 & 0 & -0.05 & -0.02 & 0.09 & -0.20 & 0 & -0.08 & -0.28 & -0.18 & 0.002 & 0 & -0.09 & -0.02 & 0.01 \\
-0.03 & 0.05 & 0 & 0.05 & 0.05 & -0.02 & 0.08 & 0 & -0.1 & -0.02 & 0.16 & 0.09 & 0 & 0.11 & 0.04 \\
-0.13 & 0.003 & -0.06 & 0 & 0.21 & 0.14 & 0.06 & 0.11 & 0 & 0.07 & 0.02 & 0.003 & -0.12 & 0 & 0.04 \\
-0.07 & -0.003 & -0.06 & -0.21 & 0 & 0.02 & 0.06 & 0.03 & 0.07 & 0 & -0.002 & -0.002 & -0.05 & -0.04 & 0 \\
0 & 0.06 & 0.14 & 0.13 & 0.07 & 0 & 0.08 & 0.09 & -0.14 & -0.02 & 0 & 0.001 & -0.68 & -0.02 & 0.002 \\
-0.06 & 0 & -0.05 & -0.003 & 0.03 & -0.08 & 0 & -0.09 & -0.06 & -0.06 & 0.001 & 0 & -0.10 & -0.003 & 0.002 \\
-0.14 & 0.05 & 0 & 0.06 & 0.06 & -0.09 & 0.09 & 0 & -0.12 & -0.03 & 0.68 & 0.10 & 0 & 0.12 & 0.05 \\
-0.21 & 0.003 & -0.03 & 0 & 0.74 & 0.23 & 0.06 & 0.07 & 0 & 0.24 & 0.04 & 0.003 & 0.08 & 0 & 0.14 \\
-0.29 & -0.03 & -0.08 & -0.74 & 0 & 0.08 & 0.07 & 0.03 & -0.24 & 0 & -0.001 & -0.003 & -0.07 & -0.13 & 0
\end{bmatrix}.$$

The rank of the matrix is 15. It means full rank and the singular values are 1.17, 0.97, 0.80, 0.76, 0.48, 0.37, 0.22, 0.14, 0.13, 0.10, 0.08, 0.06, 0.03, 0.028, 0.0026. The condition number is around 450. The singularity of the matrix is weak, and its linear constraint redundancy is low. While strict theoretical proof of the full rank remains elusive, all the linear systems in our experiments are invertible.

3 Derivation of Anisotropic Fundamental Solutions

This section will give the anisotropic fundamental solution. With the scaling matrix

$$\boldsymbol{D} = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix},$$

and our anisotropic equation is

$$\begin{aligned} &-\mathbf{div}\left(\boldsymbol{D}\cdot\nabla u\right)(x_1,x_2,x_3) \\ =&-\left(d_1\frac{\partial^2 u}{\partial x_1^2}+d_2\frac{\partial^2 u}{\partial x_2^2}+d_3\frac{\partial^2 u}{\partial x_3^2}\right)(x_1,x_2,x_3)=\delta(x_1,x_2,x_3). \end{aligned} \tag{1}$$

Let us define a function $\overline{u}(x_1,x_2,x_3)=u(\sqrt{d_1}x_1,\sqrt{d_2}x_2,\sqrt{d_3}x_3)$, it is easy to verify

$$-\Delta\overline{u}(x_1,x_2,x_3)=-\left(d_1\frac{\partial^2 u}{\partial x_1^2}+d_2\frac{\partial^2 u}{\partial x_2^2}+d_3\frac{\partial^2 u}{\partial x_3^2}\right)(\sqrt{d_1}x_1,\sqrt{d_1}x_2,\sqrt{d_1}x_3).$$

Following Equation (1), we can further get

$$-\Delta\overline{u}(x_1,x_2,x_3)=\delta(\sqrt{d_1}x_1,\sqrt{d_2}x_3,\sqrt{d_1}x_3)=\frac{1}{\sqrt{d_1d_2d_3}}\delta(x_1,x_2,x_3).$$

The classical fundamental solution of $n-$dimensions has the generalized analytical formula

$$\Phi(\boldsymbol{x})=\begin{cases} -\dfrac{1}{2\pi}\log|\boldsymbol{x}|, & n=2 \\ \dfrac{1}{n(n-2)\omega(n)}\dfrac{1}{|\boldsymbol{x}|^{n-2}}, & n\geqslant 3 \end{cases} \tag{2}$$

for the original Laplace equation. Where $\omega(n)$ is the volume of a n-dimensional unit sphere, namely $\omega(n)=\pi^{\frac{n}{2}}/\Gamma(\frac{n}{2}+1)$, and $\Gamma(\cdot)$ is the Gamma function. It means

$$\overline{u}(x_1,x_2,x_3)=\frac{1}{4\pi\sqrt{d_1d_2d_3}}\frac{1}{\sqrt{x_1^2+x_2^2+x_3^2}}.$$

Furthermore, the definition of $\overline{u}$ indicates

$$\begin{aligned} u(x_1,x_2,x_3) &= \overline{u}\left(\frac{x_1}{\sqrt{d_1}},\frac{x_2}{\sqrt{d_2}},\frac{x_3}{\sqrt{d_3}}\right) \\ &= \frac{1}{4\pi\sqrt{d_1d_2d_3}}\frac{1}{\sqrt{\frac{x_1^2}{d_1}+\frac{x_2^2}{d_2}+\frac{x_3^2}{d_3}}}. \end{aligned}$$

This is our anisotropic fundamental solution.

■

4 Derivation of Anisotropic Gauss Formula

We proof the anisotropic Gauss formula in this section. We will prove the theorem in three different situations $\boldsymbol{x}\in\Omega$, $\boldsymbol{x}\in\partial\Omega$ and $\boldsymbol{x}\in\overline{\Omega}^c$, where $\overline{\Omega}^c$ represents the complement of the closure of Ω.

First, for $\boldsymbol{x}\in\overline{\Omega}^c$, by using the divergence theorem and the fact that $\Phi_{\boldsymbol{d}}$ is smooth for $\boldsymbol{y}\in\Omega$, $\boldsymbol{x}\in\overline{\Omega}^c$, and let $z_1=x_1-y_1, z_2=x_2-y_2, z_3=x_3-y_3$ for brevity

$$\begin{aligned} &\int_{\partial\Omega} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{y})\mathrm{d}S(\boldsymbol{y}) \\ &= \int_{\partial\Omega}(\boldsymbol{D}\cdot\nabla\Phi_{\boldsymbol{d}})(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{y})\mathrm{d}S(\boldsymbol{y}) \\ &= \int_{\Omega}\mathbf{div}(\boldsymbol{D}\cdot\nabla\Phi_{\boldsymbol{d}}(\boldsymbol{x}-\boldsymbol{y}))\mathrm{d}\boldsymbol{y} \\ &= \int_{\Omega} 0\ \mathrm{d}\boldsymbol{y} \quad \text{(by Equation (1))} \\ &= 0. \end{aligned}$$

Hence, $\forall\boldsymbol{d}\in\mathbb{R}^3$, $\forall\boldsymbol{x}\in\overline{\Omega}^c$,

$$\int_{\partial\Omega} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{y})\mathrm{d}S(\boldsymbol{y})=0.$$

Secondly, for $\boldsymbol{x}\in\Omega$, $\Phi_{\boldsymbol{d}}(\boldsymbol{x}-\boldsymbol{y})$ is not smooth for at $\boldsymbol{x}$. In order to overcome this problem, we fix $\varepsilon>0$ sufficiently small such that the ellipsoid $\widetilde{B}(\boldsymbol{x};\varepsilon):=\left\{(y_1,y_2,y_3)\middle|\frac{(y_1-x_1)^2}{d_1}+\frac{(y_2-x_2)^2}{d_2}+\frac{(y_3-x_3)^2}{d_3}\leqslant\varepsilon^2\right\}$ is contained in Ω. Then in the region $\Omega-\widetilde{B}(\boldsymbol{x};\varepsilon)$, $\Phi_{\boldsymbol{d}}(\boldsymbol{x}-\boldsymbol{y})$ is smooth and we can obtain that

$$\begin{aligned} 0 &= \int_{\Omega-\widetilde{B}(\boldsymbol{x};\varepsilon)}\mathbf{div}(\boldsymbol{D}\cdot\nabla\Phi_{\boldsymbol{d}})\mathrm{d}\boldsymbol{y} \\ &= \int_{\partial\Omega-\partial\widetilde{B}(\boldsymbol{x};\varepsilon)} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{y})\mathrm{d}S(\boldsymbol{y}) \\ &= \int_{\partial\Omega} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{y})\mathrm{d}S(\boldsymbol{y}) \\ &\quad + \int_{\partial\widetilde{B}(\boldsymbol{x};\varepsilon)} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{n}_{\widetilde{B}}(\boldsymbol{y})\mathrm{d}S(\boldsymbol{y}), \end{aligned}$$

where $\boldsymbol{n}_{\widetilde{B}}$ is the inner unit normal to $\widetilde{B}(\boldsymbol{x};\varepsilon)$, which is given by

$$\boldsymbol{n}_{\widetilde{B}}(\boldsymbol{y})=\frac{\left(\frac{2(x_1-y_1)}{d_1},\frac{2(x_2-y_2)}{d_2},\frac{2(x_3-y_3)}{d_3}\right)}{\left|\left(\frac{2(x_1-y_1)}{d_1},\frac{2(x_2-y_2)}{d_2},\frac{2(x_3-y_3)}{d_3}\right)\right|},$$

for $\boldsymbol{y}\in\partial\widetilde{B}(\boldsymbol{x};\varepsilon)$.

For $\Sigma: F(x,y,z)=0$, we have

$$\int_{\Sigma_{x,y,z}} f(x,y,z)\mathrm{d}S(x,y,z)=\int_{\Sigma_{u,v,w}} g(u,v,w)\frac{|J||D_F|_2}{\left|J\cdot D_F^T\right|_2}\mathrm{d}S(u,v,w), \tag{3}$$

where: $g(u,v,w) = f(x(u,v,w), y(u,v,w), z(u,v,w))$, $D_F = (F_x, F_y, F_z)$, and $|J| = \left|\dfrac{\partial(x,y,z)}{\partial(u,v,w)}\right|$ is the absolute value of Jacobian determinant, in detail:

$$J = \begin{bmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{bmatrix}.$$

Denote

$$\begin{aligned} F(\boldsymbol{d},\varepsilon) &= \int_{\partial \widetilde{B}(\boldsymbol{x};\varepsilon)} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{n}_{\widetilde{B}}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}) \\ &= \int_{\partial \widetilde{B}(\boldsymbol{x};\varepsilon)} \frac{1}{4\pi\sqrt{d_1 d_2 d_3}} \frac{(z_1,z_2,z_3)}{\left(\frac{z_1^2}{d_1}+\frac{z_2^2}{d_2}+\frac{z_3^2}{d_3}\right)^{\frac{3}{2}}} \cdot \frac{\left(\frac{2z_1}{d_1},\frac{2z_2}{d_2},\frac{2z_3}{d_3}\right)}{\left|\left(\frac{2z_1}{d_1},\frac{2z_2}{d_2},\frac{2z_3}{d_3}\right)\right|} \mathrm{d}S(\boldsymbol{y}), \end{aligned}$$

where $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$, and $\dfrac{z_1^2}{d_1} + \dfrac{z_2^2}{d_2} + \dfrac{z_3^2}{d_3} = \varepsilon^2$. Then $\forall \boldsymbol{d} \in \mathbb{R}^3$, we take the following linear transformation:

$$\begin{cases} z_1 = \sqrt{d_1}\overline{z}_1 \\ z_2 = \sqrt{d_2}\overline{z}_2 \\ z_3 = \sqrt{d_3}\overline{z}_3 \end{cases}.$$

Then,

$$\begin{aligned} F(\boldsymbol{d},\varepsilon) = &\int_{\partial B(\boldsymbol{x};\varepsilon)} \frac{-1}{4\pi\sqrt{d_1 d_2 d_3}} \frac{\left(\sqrt{d_1}\overline{z}_1, \sqrt{d_2}\overline{z}_2, \sqrt{d_3}\overline{z}_3\right)}{\left(\overline{z}_1^2+\overline{z}_2^2+\overline{z}_3^2\right)^{\frac{3}{2}}} \cdot \frac{\left(\frac{2z_1}{d_1},\frac{2z_2}{d_2},\frac{2z_3}{d_3}\right)}{\left|\left(\frac{2z_1}{d_1},\frac{2z_2}{d_2},\frac{2z_3}{d_3}\right)\right|} \\ &\cdot \sqrt{d_1 d_2 d_3} \cdot \left|\left(\frac{2z_1}{d_1},\frac{2z_2}{d_2},\frac{2z_3}{d_3}\right)\right| \cdot \frac{1}{\left|\left(\frac{2z_1}{\sqrt{d_1}},\frac{2z_2}{\sqrt{d_2}},\frac{2z_3}{\sqrt{d_3}}\right)\right|} \mathrm{d}S(\overline{\boldsymbol{y}}) \\ &\text{(by Equation (3))} \\ = &-\int_{\partial B(\boldsymbol{x};\varepsilon)} \frac{1}{4\pi\varepsilon^2} \mathrm{d}S(\overline{\boldsymbol{y}}) \\ = &-1, \end{aligned}$$

where: $B(\boldsymbol{x};\varepsilon) = \left\{\boldsymbol{y} \middle| |\boldsymbol{y}-\boldsymbol{x}| \leqslant \varepsilon\right\}$ is a ball with the radius of ε.

In other words, $\forall \boldsymbol{d} \in \mathbb{R}^3$, $\forall \boldsymbol{x} \in \Omega$,

$$\int_{\partial\Omega} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}) = 1.$$

Lastly, we consider the case $\boldsymbol{x} \in \partial\Omega$. In this case, $\Phi_{\boldsymbol{d}}(\boldsymbol{x}-\boldsymbol{y})$ is not defined at $\boldsymbol{y} = \boldsymbol{x}$. Fix $\boldsymbol{x} \in \partial\Omega$, and let

$$\widetilde{B}(\boldsymbol{x};\varepsilon) := \left\{(y_1,y_2,y_3) \middle| \frac{(y_1-x_1)^2}{d_1} + \frac{(y_2-x_2)^2}{d_2} + \frac{(y_3-x_3)^2}{d_3} \leqslant \varepsilon^2\right\},$$

which is the ellipsoid with $\boldsymbol{x}$ as the center and $B(\boldsymbol{x};\varepsilon)$ be the ball of radius r centered at $\boldsymbol{x}$. Let $\Omega_\varepsilon \equiv \Omega - (\Omega \cap \widetilde{B}(\boldsymbol{x};\varepsilon))$, and $\mathcal{C}_\varepsilon \equiv \{\boldsymbol{y} \in \partial\widetilde{B}(\boldsymbol{x};\varepsilon) : \boldsymbol{n}(\boldsymbol{x}) \cdot (\boldsymbol{y}-\boldsymbol{x}) < 0\}$, and $\widetilde{\mathcal{C}_\varepsilon} \equiv \partial\Omega_\varepsilon \cap \mathcal{C}_\varepsilon$.

Then, we start to prove this situation. First, we note that

$$\begin{aligned} 0 &= \int_{\Omega_\varepsilon} \mathbf{div}(\boldsymbol{D} \cdot \nabla\Phi_{\boldsymbol{d}}) \mathrm{d}\boldsymbol{y} \\ &= \int_{\partial\Omega_\varepsilon} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}) \\ &= \int_{\partial\Omega_\varepsilon - \widetilde{\mathcal{C}_\varepsilon}} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}) \\ &+ \int_{\widetilde{\mathcal{C}_\varepsilon}} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{n}_{\widetilde{B}}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}), \end{aligned}$$

where $\boldsymbol{n}_{\widetilde{B}}$ is the inner unit normal to Ω_ε, and

$$\boldsymbol{n}_{\widetilde{B}}(\boldsymbol{y}) = \frac{\left(\frac{2(x_1-y_1)}{d_1}, \frac{2(x_2-y_2)}{d_2}, \frac{2(x_3-y_3)}{d_3}\right)}{\left|\left(\frac{2(x_1-y_1)}{d_1}, \frac{2(x_2-y_2)}{d_2}, \frac{2(x_3-y_3)}{d_3}\right)\right|}.$$

Since $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$, and $\frac{z_1^2}{d_1} + \frac{z_2^2}{d_2} + \frac{z_3^2}{d_3} = \varepsilon^2$. Then $\forall \boldsymbol{d} \in \mathbb{R}^3$, we also make the following linear transformation:

$$\begin{cases} z_1 = \sqrt{d_1}\overline{z}_1 \\ z_2 = \sqrt{d_2}\overline{z}_2 \\ z_3 = \sqrt{d_3}\overline{z}_3 \end{cases}.$$

Then, in the transformed coordinate system, denote $\overline{\mathcal{C}}_\varepsilon \equiv \{\boldsymbol{y} \in \partial B(\boldsymbol{x};\varepsilon) : \boldsymbol{n}\cdot\boldsymbol{y} < 0\}$, and $\widetilde{\overline{\mathcal{C}}}_\varepsilon \equiv \partial\Omega_\varepsilon \cap \overline{\mathcal{C}}_\varepsilon$. In addition, we provide a lemma (Equation (4)) that needs to be used and provide proof of it.

For $\widetilde{\overline{\mathcal{C}}}_\varepsilon$ and $\overline{\mathcal{C}}_\varepsilon$ as defined above, we have

$$\int_{\widetilde{\overline{\mathcal{C}}}_\varepsilon} \mathrm{d}S(\boldsymbol{y}) = \int_{\overline{\mathcal{C}}_\varepsilon} \mathrm{d}S(\boldsymbol{y}) + O(\varepsilon^3). \tag{4}$$

Proof. We just need to show that the surface area of $\widetilde{\overline{\mathcal{C}}}_\varepsilon - \overline{\mathcal{C}}_\varepsilon$ is $O(\varepsilon^3)$. The surface area is approximately the surface area of the base times the height. Now the surface area of the base is $O(\varepsilon)$. Therefore, we just need to show that the height is $O(\varepsilon^2)$.

Without loss of generality, we let $\boldsymbol{x} = \boldsymbol{0}$. Now, by assumption, $\partial\Omega$ is C^2. Therefore, $\partial\Omega$ can be written as the graph of a C^2 function f: $\mathbb{R}^2 \to \mathbb{R}$ such that $f(\boldsymbol{0}) = 0$ and $\nabla f(\boldsymbol{0}) = 0$. Therefore, if $\boldsymbol{y} = (y_1, y_2, y_3) \in \overline{\mathcal{C}}_\varepsilon - \widetilde{\overline{\mathcal{C}}}_\varepsilon$, then

$$|y_3| \leqslant |f(y_1, y_2)| \leqslant C|(y_1, y_2)|^2 \leqslant C\varepsilon^2,$$

by using Taylor's theorem. Therefore, the height is $O(\varepsilon^2)$ and the Equation (4) follows. □

Then, we can get

$$\begin{aligned}
F_{\partial\Omega}(\boldsymbol{d},\varepsilon) &= \int_{\widetilde{\mathcal{C}}_\varepsilon} K_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{n}_{\widetilde{B}}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}) \\
&= -\int_{\widetilde{\overline{\mathcal{C}}}_\varepsilon} \frac{1}{4\pi\sqrt{d_1 d_2 d_3}} \frac{\left(\sqrt{d_1}\overline{z}_1, \sqrt{d_2}\overline{z}_2, \sqrt{d_3}\overline{z}_3\right)}{\left(\overline{z}_1^2 + \overline{z}_2^2 + \overline{z}_3^2\right)^{\frac{3}{2}}} \cdot \frac{\left(\frac{2z_1}{d_1}, \frac{2z_2}{d_2}, \frac{2z_3}{d_3}\right)}{\left|\left(\frac{2z_1}{d_1}, \frac{2z_2}{d_2}, \frac{2z_3}{d_3}\right)\right|} \\
&\cdot \sqrt{d_1 d_2 d_3} \cdot \left|\left(\frac{2z_1}{d_1}, \frac{2z_2}{d_2}, \frac{2z_3}{d_3}\right)\right| \cdot \frac{1}{\left|\left(\frac{2z_1}{\sqrt{d_1}}, \frac{2z_2}{\sqrt{d_2}}, \frac{2z_3}{\sqrt{d_3}}\right)\right|} \mathrm{d}S(\boldsymbol{y}) \\
&\text{(by Equation (3))} \\
&= -\int_{\widetilde{\overline{\mathcal{C}}}_\varepsilon} \frac{1}{4\pi\varepsilon^2} \mathrm{d}S(\boldsymbol{y}) \\
&= -\frac{1}{4\pi\varepsilon^2} \cdot \left(\int_{\mathcal{C}_\varepsilon} \mathrm{d}S(\boldsymbol{y}) + O(\varepsilon^3)\right) \quad \text{(by Equation (4))} \\
&= -\frac{1}{2} + O(\varepsilon) \triangleq F(\boldsymbol{d},\varepsilon).
\end{aligned}$$

Taking the limit as $\varepsilon \to 0^+$, then

$$\lim_{\varepsilon\to 0^+} F(\boldsymbol{d},\varepsilon) = -\frac{1}{2}.$$

In other words, $\forall \boldsymbol{d} \in \mathbb{R}^3$, $\forall \boldsymbol{x} \in \partial\Omega$,

$$\int_{\partial\Omega} K_{\boldsymbol{d}}(\boldsymbol{x}-\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{y}) \mathrm{d}S(\boldsymbol{y}) = \frac{1}{2}.$$

■

5 Comparison of the Running Speed for Our Method and Other State-of-the-art Methods

Table 2: The quantitative comparison of running time for our method and the main part of other state-of-the-art methods.

Time(/s)	1K	5K	10K	50K	500K	1M	5M
GCNO	48.7	876.4	4582.4	-	-	-	-
iPSR	10.2	23.6	35.5	65.5	1487.6	2866.8	4834.5
PGR	1.7	5.8	20.5	115.4	-	-	-
WNNC	0.6	0.8	1.6	8.6	13.9	36.4	132.9
Ours	0.7	0.9	1.9	9.4	15.6	45.5	157.8

Running speed serves as a crucial metric for evaluating algorithm quality. In our paper (*"Anisotropic Gauss Reconstruction and Global Orientation with Octree-based Acceleration"*), we show the qualitative evaluation and comparison of the computational efficiency across different state-of-the-art methods. In this section, we provide detailed algorithm computation times as quantitative evidence to emphasize our significant contribution and outstanding performance in octree-based acceleration.

As shown in Table 2, our algorithm can complete the reconstruction of point clouds below 500K within a few seconds on an NVIDIA GeForce 4090 GPU with 24GB memory.

In conclusion, we will emphasize the core innovation of our method, which involves introducing anisotropy by scaling vectors, deriving the anisotropic Gauss formula, and developing the octree-based acceleration algorithms. Our method demonstrates the ability to handle difficult point clouds, especially with small holes, thin structures, or noise. Extensive experiments demonstrate that our method achieves state-of-the-art performance in orientation and surface reconstruction for unoriented point clouds. Meanwhile, our method has more solid theoretical support in terms of convergence and can handle large-scale point clouds, reducing space complexity to $O(N)$ and time complexity to $O(N \log N)$.