

A. Additional Experiments

A.1. Detailed description of Shapenet Experiment

Many of the methods we compared against on Shapenet have tunable parameters which can drastically alter the quality of reconstructed outputs. To ensure a proper comparison, we ran sweeps over these parameters where appropriate choosing the best reconstruction for each model under both metrics (Chamfer and IoU). For our method we did no parameter sweeps, using no regularization and 1024 Nyström samples for each model in the dataset. We describe the experimental methodology for each method in the benchmark below.

Implicit Geometric Regularization [24] We trained each model for 5k iterations with Adam and a learning rate of 0.001 using the same parameters and architecture as proposed in the original paper. We included the normals in the loss with the parameter τ set to 1. The Eikonal regularization term λ was set to 0.1. While IGR can slightly improve by using a very large number of iterations, doing so is prohibitively slow over many models. Figure 8 motivates our choice of iterations, demonstrating only a slight improvements between 5k and 100k iterations (the latter which required 2 hours of fitting on a NVIDIA-1080-Ti GPU).

Screened Poisson Surface Reconstruction [31] We considered every possible combination of the following parameters: the *octree depth* in [6, 7, 8, 9], the number of *points per leaf* in [1, 2, 3, 5, 10], the *point weight* (which controls the degree to which the method interpolates the input) in [4.0, 100.0, 1000.0]. Since all the shapes in the benchmark are watertight meshes, we used Dirichlet boundary constraints for the reconstruction.

SIREN [40] We trained used a SIREN network with 4 hidden layers each containing 256 neurons for 5000 iterations, using a learning rate of 1e-4 using Adam. We used a loss function which encouraged samples at the points to have zero value, and a L2 loss on the gradient of the function and the normals at the points. We trained both with and without an eikonal regularization term ($(\|\nabla f(x)\| - 1)^2$) and found that results improved with the eikonal term.

Fourier Feature Networks [42] We used an 8-layer ReLU MLP with 256 Fourier features sampled from a Gaussian distribution. This is the same architecture as the shape representation experiment in the original paper. For each model in the benchmark, we did a parameter sweep on the variance σ of the Gaussian distribution, considering $\sigma \in \{0.1, 0.25, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25, 1.5, 3.0\}$. The range of parameters was chosen by empirical verification on 3 models from the airplanes, benches, and cars categories.

SVR [22] As in the original paper, we use a Gaussian kernel to perform support vector regression. To generate occupancy samples, we augmented the input points with an "inside" and "outside" point by perturbing them by $\pm\epsilon$ along the normal at that point. We used $\epsilon = 0.01$ for all the models. For each model we did a joint parameter sweep over the regularization parameter $C \in \{1.0, 0.1, 0.01, 0.001, 0.0001\}$ and the variance parameter $\sigma \in \{0.002, 0.001, 0.0004, 0.0002, 0.0001\}$. The range of parameters was chosen by empirical verification on 3 models from the airplanes, benches, and cars categories.

Biharmonic RBF [9] To generate occupancy samples, we augmented the input points with an "inside" and "outside" point by perturbing them by $\pm\epsilon$ along the normal at that point. We used $\epsilon = 0.01$ for all the models. The biharmonic function is very simple $\phi(r) = r$ where $r = \|x_i - x_j\|$ and does not require tuning parameters.

A.2. Detailed Description of Surface Reconstruction Benchmark Experiment

For the surface reconstruction benchmark we only included comparisons against neural network based methods since [44] performed an extensive comparison against traditional methods and clearly established itself as superior. We used the same experimental setup and parameter sweeps as in the Shapenet benchmark (Section A.1) for IGR, SIREN and Fourier feature networks. We verified that the range of parameters used for Fourier feature networks was valid by qualitative verification on the models (there are only 5 so this is straightforward). For our method we use 15000 Nyström samples. Additionally, since noise is present in the input data, we performed a modest parameter sweep over regularization parameters considering $\lambda \in \{0, 1e-13, 1e-12, 1e-11, 1e-10\}$. We remark that a full parameter sweep with our method requires less time than fitting a single model with competing methods (See timings in Section A.5).

A.3. Additional Figures

Figures 10 and 11 show at least one model reconstructed from each Shapenet category using our method, Implicit Geometric Regularization (IGR) [24], SIREN [40], Fourier Feature Networks (FFN) [42], Screened Poisson Surface Reconstruction [31], Biharmonic RBF [9], and Support Vector Regression (SVR) [22]. Figure 12 shows the reconstructions of all the models from the Surface Reconstruction Benchmark [5] using our method, IGR [24], SIREN [40], and Fourier Feature Networks [42].

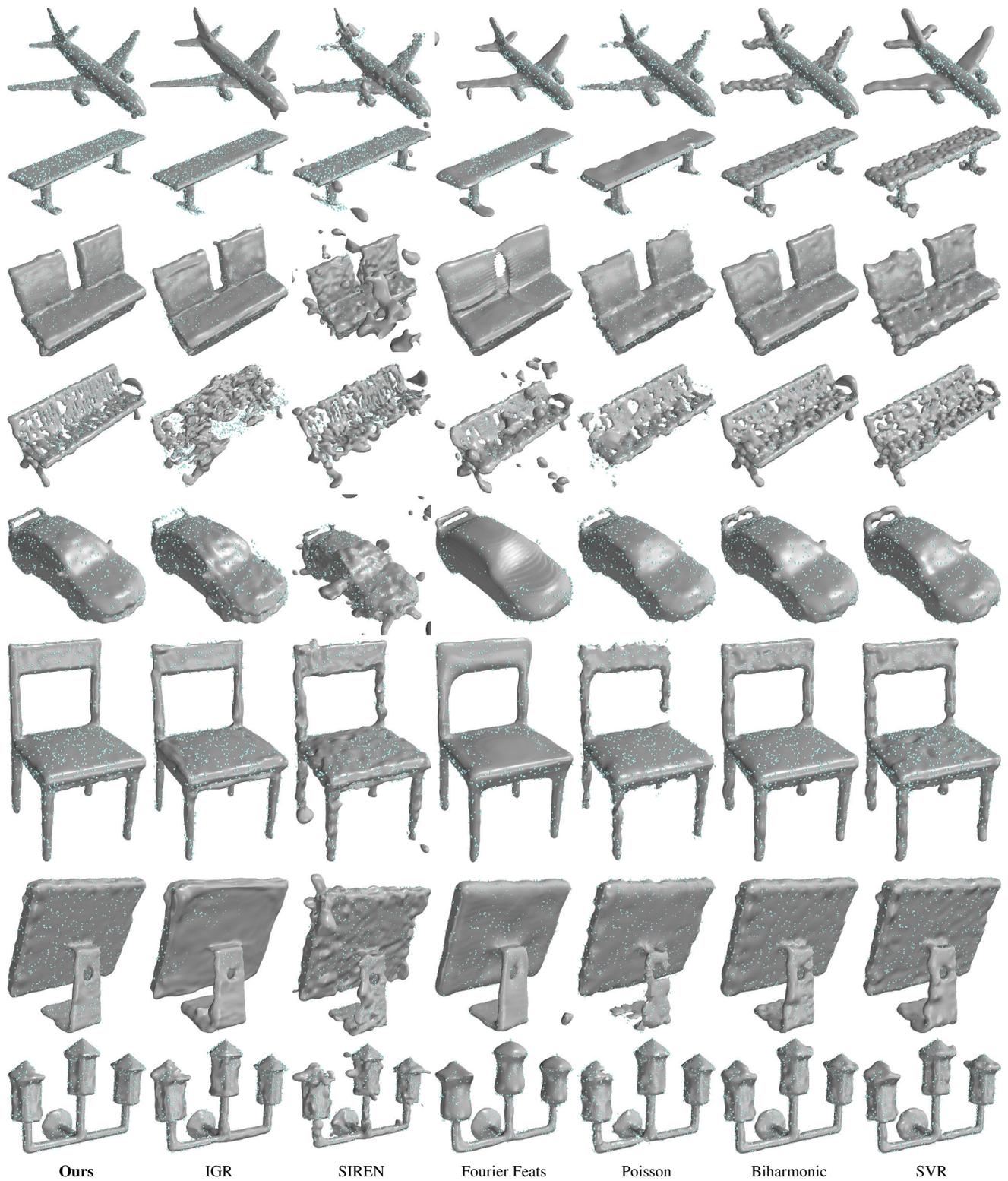


Figure 10. Comparisons between reconstruction techniques on Shapenet models. The blue points are the input points to the reconstruction algorithm.

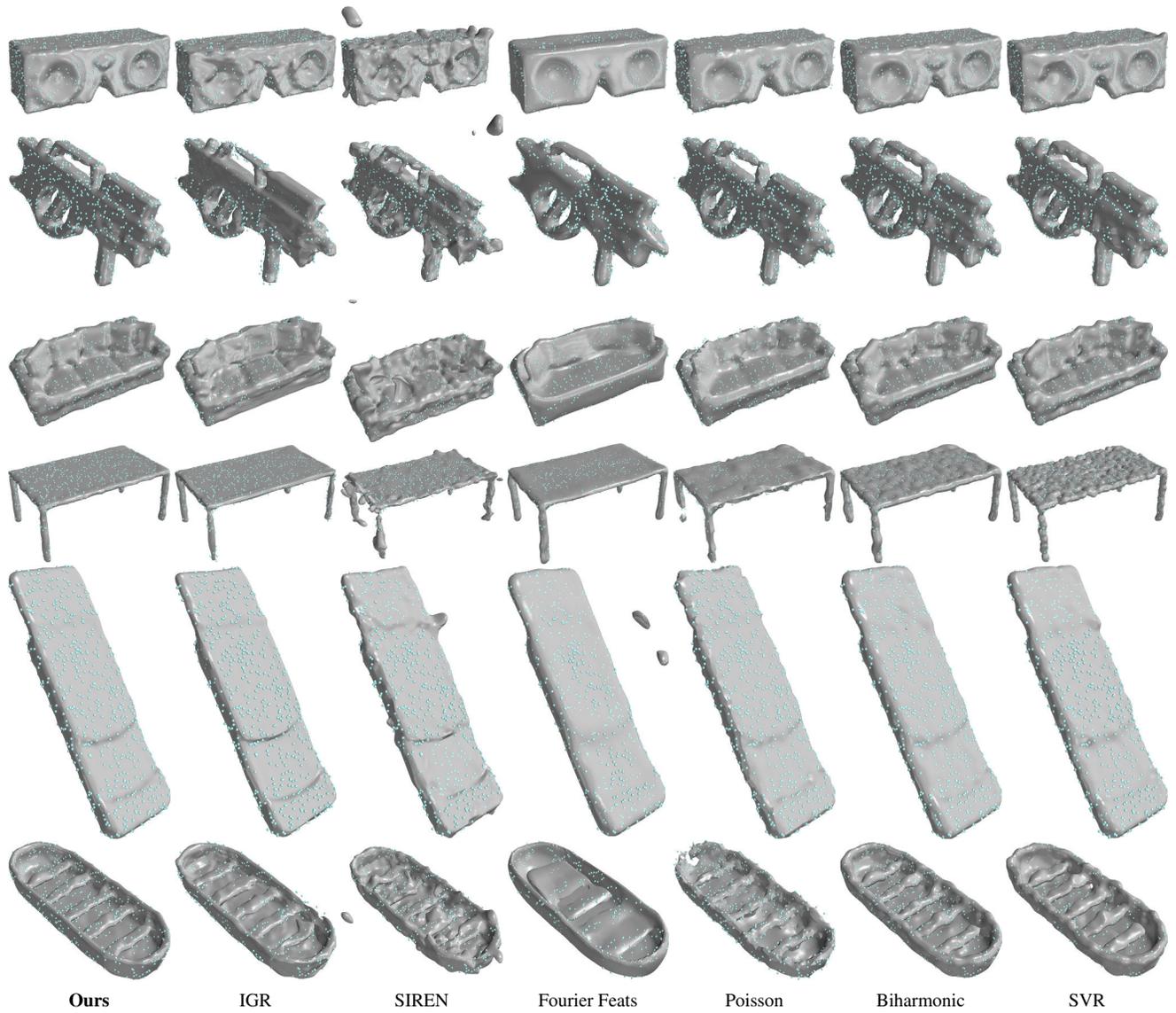
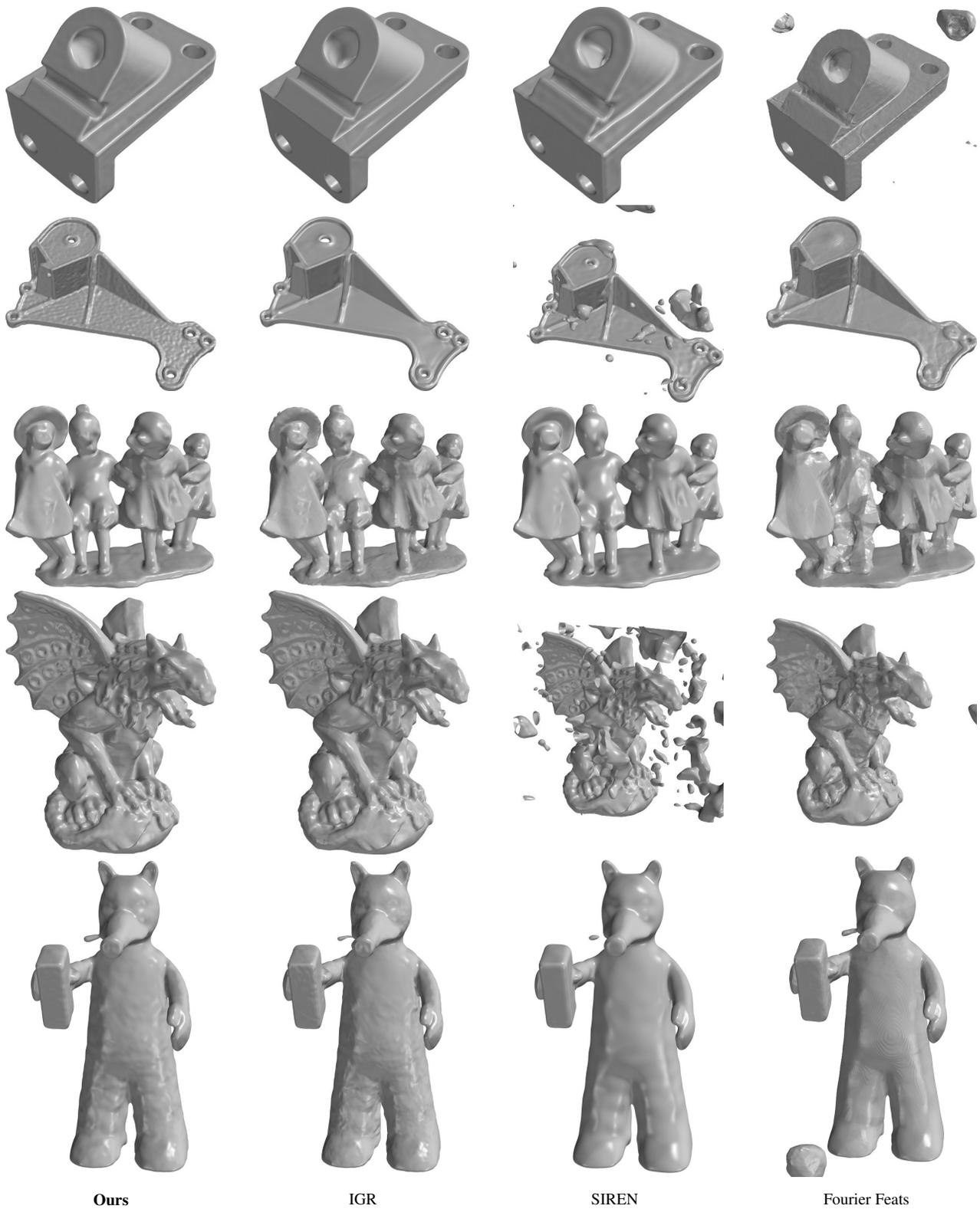


Figure 11. Comparisons between reconstruction techniques on Shapenet models. The blue points are the input points to the reconstruction algorithm.



Ours

IGR

SIREN

Fourier Feats

Figure 12. Comparisons between reconstruction techniques on the Surface Reconstruction Benchmark models. For techniques requiring parameter sweeps, we show the result with the lowest Chamfer Distance.

A.4. Quantitative Results Per ShapeNet Class

Tables 4 show the per ShapeNet category IoU and Chamfer distance statistics for the benchmark described in Section 4.1.

Intersection over Union (IoU)

Class	SIREN [40]			Fourier Feat. Nets [42]			Biharmonic RBF [9]			SVR [22]			Screened Poisson [31]			IGR [24]			Ours		
	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std
car	0.7331	0.7385	0.1134	0.8106	0.8238	0.0858	0.8656	0.9020	0.1004	0.7903	0.8065	0.0773	0.6637	0.7192	0.1255	0.8102	0.8459	0.1236	0.9082	0.9399	0.0747
chair	0.8143	0.8304	0.0665	0.8202	0.8450	0.0675	0.8428	0.8667	0.0824	0.8127	0.8389	0.0839	0.5880	0.6046	0.1243	0.8216	0.8654	0.1139	0.9056	0.9369	0.1062
airplane	0.7410	0.7572	0.0764	0.6811	0.6970	0.0481	0.6690	0.7064	0.0809	0.6016	0.6257	0.0611	0.5954	0.6139	0.0581	0.7804	0.8161	0.0913	0.7773	0.8796	0.1191
display	0.8512	0.8663	0.0531	0.8651	0.8669	0.0323	0.8470	0.8593	0.0738	0.8307	0.8407	0.0722	0.7027	0.7072	0.0880	0.8673	0.8990	0.0690	0.9533	0.9549	0.0188
table	0.7562	0.7535	0.0787	0.7776	0.7751	0.0688	0.7096	0.7350	0.1150	0.6779	0.6596	0.1127	0.3720	0.3565	0.1334	0.7747	0.7772	0.0742	0.8968	0.9011	0.0500
table	0.8088	0.8442	0.0840	0.7595	0.7666	0.0726	0.8350	0.8447	0.0845	0.7764	0.8013	0.0970	0.6803	0.6791	0.0532	0.8381	0.8479	0.0568	0.9489	0.9491	0.0169
cabinet	0.8592	0.8687	0.0813	0.8885	0.8922	0.0686	0.8913	0.9206	0.0926	0.7268	0.7142	0.0740	0.7301	0.7764	0.1095	0.8853	0.8916	0.0782	0.9478	0.9467	0.0377
loudspeaker	0.8681	0.8994	0.0873	0.8900	0.9226	0.0758	0.9076	0.9621	0.1067	0.7098	0.6956	0.1193	0.7432	0.7848	0.1309	0.8595	0.9452	0.2038	0.9507	0.9768	0.0518
telephone	0.9088	0.9183	0.0536	0.9104	0.9156	0.0461	0.9286	0.9433	0.0662	0.9195	0.9399	0.0638	0.7883	0.7990	0.0552	0.9148	0.9372	0.0639	0.9746	0.9772	0.0202
bench	0.6146	0.6884	0.1853	0.6685	0.6765	0.1488	0.6237	0.6592	0.1082	0.6052	0.6282	0.1487	0.4728	0.4384	0.1213	0.5862	0.6394	0.2486	0.8160	0.8900	0.1365
sofa	0.8704	0.8954	0.0589	0.8811	0.9043	0.0553	0.9164	0.9377	0.0630	0.8694	0.8791	0.0740	0.7122	0.7326	0.0662	0.8870	0.9210	0.1211	0.9565	0.9644	0.0267
watercraft	0.7600	0.7989	0.1147	0.8160	0.8187	0.0570	0.8557	0.8756	0.0768	0.8197	0.8379	0.0742	0.6823	0.6793	0.0994	0.8272	0.8537	0.0845	0.9340	0.9380	0.0462
lamp	0.8104	0.8143	0.0833	0.8178	0.8334	0.0729	0.8282	0.8706	0.1155	0.7728	0.7907	0.0963	0.5413	0.5786	0.1858	0.8251	0.8352	0.1093	0.9467	0.9470	0.0306
All Classes:	0.7997	0.8248	0.1203	0.8143	0.8321	0.1047	0.8247	0.8642	0.1350	0.7625	0.7819	0.1300	0.6340	0.6728	0.1577	0.8213	0.8566	0.1461	0.9167	0.9438	0.0985

Chamfer Distance (CD)

Class	SIREN [40]			Fourier Feat. Nets [42]			Biharmonic RBF [9]			SVR [22]			Screened Poisson [31]			IGR [24]			Ours		
	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std
car	1.54e-4	1.53e-4	4.10e-5	1.20e-4	1.13e-4	3.10e-5	1.17e-4	1.00e-4	5.23e-5	1.23e-4	1.18e-4	3.90e-5	2.27e-4	2.22e-4	8.04e-5	2.60e-4	2.82e-4	9.80e-5	8.21e-5	7.23e-5	3.59e-5
chair	1.04e-4	8.55e-5	4.64e-5	1.01e-4	8.82e-5	4.25e-5	1.10e-4	9.60e-5	6.11e-5	1.11e-4	1.01e-4	5.66e-5	2.82e-4	2.13e-4	1.84e-4	9.25e-4	9.88e-5	3.11e-3	5.52e-5	4.20e-5	4.01e-5
airplane	9.96e-5	8.22e-5	6.44e-5	1.03e-4	1.07e-4	1.24e-5	1.41e-4	1.31e-4	3.07e-5	1.61e-4	1.66e-4	1.01e-5	8.37e-5	8.76e-5	1.77e-5	3.04e-4	1.74e-4	3.47e-4	3.52e-5	3.45e-5	2.45e-6
display	7.96e-5	7.94e-5	2.46e-5	7.76e-5	7.42e-5	2.45e-5	9.85e-5	9.06e-5	3.42e-5	9.86e-5	1.01e-4	3.35e-5	2.45e-4	2.13e-4	1.11e-4	9.99e-5	7.49e-5	8.44e-5	4.31e-5	3.95e-5	1.24e-5
table	1.06e-4	9.34e-5	4.62e-5	1.03e-4	9.49e-5	3.74e-5	1.96e-4	1.91e-4	8.62e-5	1.66e-4	1.60e-4	6.73e-5	3.50e-4	2.53e-4	2.14e-4	3.40e-4	1.95e-4	3.33e-4	6.44e-5	4.78e-5	4.01e-5
table	5.10e-5	4.69e-5	1.36e-5	7.10e-5	6.67e-5	1.66e-5	5.04e-5	5.00e-5	9.36e-6	7.20e-5	7.12e-5	1.46e-5	4.46e-5	3.11e-5	2.76e-5	9.62e-5	5.29e-5	1.25e-4	3.27e-5	3.16e-5	2.81e-6
cabinet	1.19e-4	9.30e-5	5.53e-5	1.04e-4	8.76e-5	3.93e-5	1.29e-4	1.19e-4	7.81e-5	1.34e-4	1.30e-4	8.27e-5	3.62e-4	2.91e-4	1.96e-4	1.56e-4	9.39e-5	1.23e-4	6.93e-5	4.67e-5	4.30e-5
loudspeaker	1.31e-4	1.00e-4	7.97e-5	1.08e-4	9.11e-5	5.48e-5	1.36e-4	7.94e-5	1.16e-4	1.36e-4	9.91e-5	1.02e-4	4.29e-4	3.54e-4	2.68e-4	3.77e-3	1.15e-4	1.49e-2	8.27e-5	4.58e-5	7.24e-5
telephone	5.61e-5	4.77e-5	1.86e-5	4.89e-5	4.55e-5	1.20e-5	4.90e-5	4.04e-5	2.92e-5	4.77e-5	4.06e-5	2.19e-5	1.28e-4	1.17e-4	3.36e-5	1.03e-4	4.43e-5	1.54e-4	3.32e-5	3.18e-5	3.42e-6
bench	1.28e-4	1.27e-4	5.44e-5	9.38e-5	9.76e-5	2.26e-5	1.58e-4	1.73e-4	4.91e-5	1.34e-4	1.41e-4	3.05e-5	1.93e-4	1.93e-4	8.14e-5	4.48e-4	2.58e-4	4.33e-4	5.62e-5	4.74e-5	2.05e-5
sofa	8.99e-5	8.52e-5	1.97e-5	9.69e-5	9.74e-5	2.14e-5	8.42e-5	7.72e-5	3.17e-5	9.46e-5	9.02e-5	3.39e-5	2.72e-4	2.46e-4	1.02e-4	2.86e-4	1.02e-4	5.30e-4	5.08e-5	4.81e-5	1.22e-5
watercraft	8.71e-5	7.93e-5	3.31e-5	8.89e-5	8.48e-5	2.00e-5	7.81e-5	6.73e-5	3.89e-5	9.47e-5	8.56e-5	3.07e-5	1.07e-4	8.82e-5	6.12e-5	1.47e-4	1.12e-4	1.23e-4	4.41e-5	3.84e-5	1.42e-5
lamp	1.05e-4	9.05e-5	5.19e-5	7.96e-5	8.33e-5	1.53e-5	9.26e-5	8.05e-5	4.81e-5	1.10e-4	1.02e-4	4.26e-5	1.61e-4	1.35e-4	1.02e-4	1.72e-3	1.28e-4	6.24e-3	4.15e-5	3.89e-5	9.61e-6
All Classes:	1.01e-4	8.62e-5	5.40e-5	9.19e-5	8.68e-5	3.47e-5	1.11e-4	8.97e-5	7.06e-5	1.14e-4	1.04e-4	5.99e-5	2.22e-4	1.70e-4	1.76e-4	6.66e-4	1.07e-4	4.69e-3	5.32e-5	4.07e-5	3.53e-5

Table 4. Quantitative comparison of the Intersection over Union (IoU) Distance and Chamfer Distance (CD) between SIREN [40], Biharmonic RBF [9], Implicit Kernel SVR [22], Screened Poisson Surface Reconstruct [31], IGR [24] and our method over a subset (20 models per class) of the ShapeNet dataset. For each method we did a sweep over a range of parameters choosing the best result for each metric. We did no such tuning for our method.

A.5. Per Model Performance Numbers

Table 5 shows the runtime and GPU usage required to reconstruct each model in the Surface Reconstruction Benchmark. For our model, we used 15k Nyström samples and a regularization of $1e-11$. We do not report CPU memory usage since it is hard to profile exactly, however we observed that none of the methods used more than 4GiB of CPU memory. All timings were done on a machine with a single NVIDIA-V100 GPU with 16GiB of VRAM, 32GiB of CPU RAM, and an 8 core Intel Xeon processor.

	Method	Runtime (seconds)	GPU Memory
Anchor	IGR [24]	1726.84	5093
	SIREN [40]	183.60	1855
	FFN [42]	385.90	4266
	Poisson [31]	2.35	N.A.
	Ours	13.22	5528
Daratech	IGR [24]	1239.43	5093
	SIREN [40]	135.46	1855
	FFN [42]	281.61	4266
	Poisson [31]	1.83	N.A.
	Ours	10.75	5528
DC	IGR [24]	1443.65	5093
	SIREN [40]	155.91	1545
	FFN [42]	325.10	3565
	Poisson [31]	1.41	N.A.
	Ours	12.31	5510
Gargoyle	IGR [24]	1646.44	5093
	SIREN [40]	203.71	2076
	FFN [42]	432.59	4778
	Poisson [31]	1.44	N.A.
	Ours	12.87	5145
Lord Quas	IGR [24]	1013.33	5093
	SIREN [40]	128.38	1237
	FFN [42]	263.34	2870
	Poisson [31]	1.21	N.A.
	Ours	10.41	5281

Table 5. Runtime and GPU memory usage of different methods when reconstructing models from the Surface Reconstruction Benchmark [5].

A.6. Empirical versus Analytical Kernel

Figure 14 compares results using the empirical kernel with m neurons and using the analytical kernel. Figure 13 shows the convergence of the empirical Kernel to the analytic one as the number m of neurons grows.

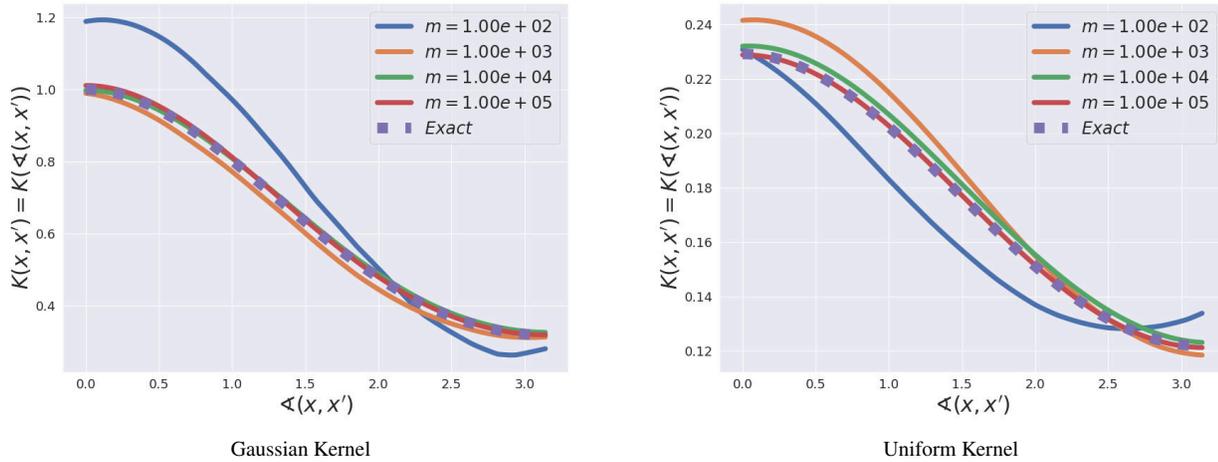


Figure 13. Convergence of the empirical kernels to the exact ones. Both the Gaussian (left) and Uniform (right) kernels are rotation invariant and thus depend only on the angle $\angle(x, x')$ between x and x' . This plot shows the value of the kernel as a function of this angle from 0 to π (we show here only the scalar term of the kernel $\mathbb{E}_{(a,b)}[ax + b]_+[ax' + b]_+$).

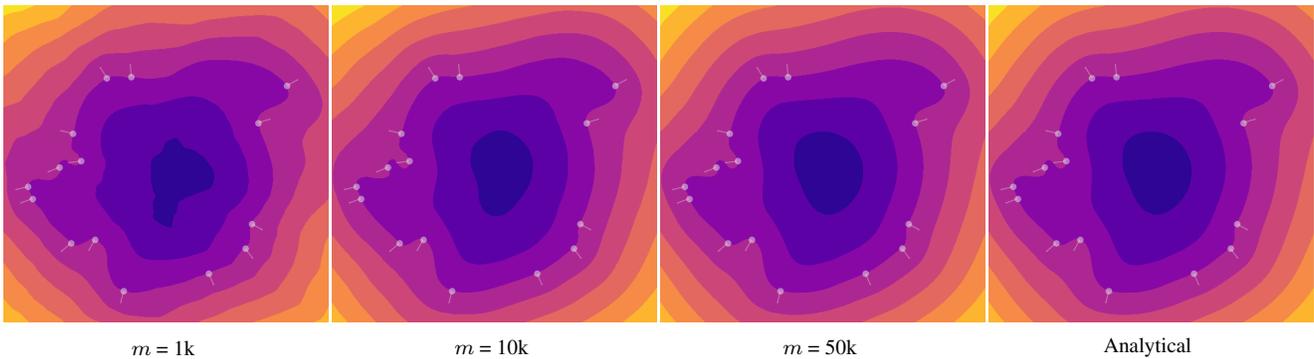


Figure 14. The effect of using an approximate kernel with m neurons to do reconstruction. Increasing m makes the approximation closer to the analytical version.

A.7. Quantitative Comparison between Gaussian and Uniform Kernels

Table 6 shows a quantitative comparison between Neural Splines using the Gaussian initialization (9) and the Uniform initialization (8) on the benchmark described in Section (4.1). The results in both cases are very close to each other in both Chamfer distance and in IoU.

Intersection over Union (IoU)

Class	Neural Spline (Uniform)			Neural Spline (Gaussian)		
	mean	median	std	mean	median	std
car	0.9082	0.9399	0.0747	0.9084	0.9392	0.0743
chair	0.9056	0.9369	0.1062	0.9131	0.9357	0.0852
airplane	0.7773	0.8796	0.1961	0.8388	0.9159	0.1531
display	0.9533	0.9549	0.0188	0.9523	0.9535	0.0193
table	0.8968	0.9011	0.0500	0.8952	0.8980	0.0508
rifle	0.9489	0.9491	0.0169	0.9488	0.9498	0.0185
cabinet	0.9478	0.9467	0.0377	0.9477	0.9475	0.0377
loudspeaker	0.9507	0.9768	0.0518	0.9500	0.9768	0.0531
telephone	0.9746	0.9772	0.0202	0.9741	0.9774	0.0209
bench	0.8160	0.8900	0.1365	0.8195	0.8902	0.1296
sofa	0.9565	0.9644	0.0267	0.9584	0.9643	0.0221
watercraft	0.9340	0.9380	0.0462	0.9360	0.9386	0.0404
lamp	0.9467	0.9470	0.0306	0.9457	0.9459	0.0319
All Classes:	0.9167	0.9438	0.0985	0.9221	0.9441	0.0834

Chamfer Distance (CD)

Class	Neural Spline (Uniform)			Neural Spline (Gaussian)		
	mean	median	std	mean	median	std
car	8.21e-05	7.23e-05	3.59e-05	8.21e-05	7.18e-05	3.60e-05
chair	5.52e-05	4.20e-05	4.01e-05	5.62e-05	4.21e-05	4.32e-05
airplane	3.55e-05	3.45e-05	2.45e-06	3.55e-05	3.44e-05	2.45e-06
display	4.31e-05	3.95e-05	1.24e-05	4.36e-05	3.99e-05	1.28e-05
table	6.44e-05	4.78e-05	4.01e-05	6.60e-05	4.88e-05	4.17e-05
rifle	3.27e-05	3.16e-05	2.81e-06	3.26e-05	3.15e-05	2.79e-06
cabinet	6.93e-05	4.67e-05	4.30e-05	6.98e-05	4.69e-05	4.34e-05
loudspeaker	8.27e-05	4.58e-05	7.24e-05	8.41e-05	4.54e-05	7.54e-05
telephone	3.33e-05	3.18e-05	3.42e-06	3.34e-05	3.19e-05	3.60e-06
bench	5.62e-05	4.74e-05	2.05e-05	5.66e-05	4.82e-05	2.09e-05
sofa	5.08e-05	4.81e-05	1.22e-05	5.11e-05	4.80e-05	1.24e-05
watercraft	4.41e-05	3.84e-05	1.42e-05	4.41e-05	3.84e-05	1.42e-05
lamp	4.15e-05	3.89e-05	9.61e-06	4.19e-05	3.91e-05	1.00e-05
All Classes:	5.32e-05	4.07e-05	3.53e-05	5.36e-05	4.06e-05	3.64e-05

Table 6. Comparison of IoU and Chamfer Distance between Neural Splines with Gaussian and Uniform kernels on the benchmark described in Section 4.1. We remark that both kernels yield extremely close results.

B. Derivation of the Infinite Width Kernels

B.1. Uniform Initialization

We derive an explicit expression for the kernel K_∞ in (5) in the case of uniform initialization (8). We first prove the following Lemma which we will use in our calculations.

Lemma 5. Assume that $\mathcal{F} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is such that $\mathcal{F}(ks, kt) = k^r \mathcal{F}(s, t)$ for some r and any $k \geq 0$. For any $x, x' \in \mathbb{R}^d$

we have that

$$\begin{aligned}
\int_{a \in \mathbb{S}^{d-1}} \mathcal{F}(a^T x, a^T x') d\Omega &= \eta_{d,r} \cdot \frac{r!!}{(d+r-2)!!} \cdot F_1 \\
\int_{a \in \mathbb{S}^{d-1}} a \mathcal{F}(a^T x, a^T x') d\Omega &= \eta_{d,r} \cdot \frac{(r+1)!!}{(d+r-1)!!} \cdot Q^T \begin{bmatrix} 0_{d-2} \\ F_{\cos} \\ F_{\sin} \end{bmatrix} \\
\int_{a \in \mathbb{S}^{d-1}} aa^T \mathcal{F}(a^T x, a^T x') d\Omega &= \eta_{d,r} \cdot \frac{(r+2)!!}{(d+r)!!} \cdot Q^T \begin{bmatrix} \frac{1}{r+2} Id_{d-2} F & 0 & 0 \\ 0 & F_{\cos^2} & F_{\sin \cos} \\ 0 & F_{\sin \cos} & F_{\sin^2} \end{bmatrix} Q
\end{aligned} \tag{17}$$

where $F_g = \int_0^{2\pi} g(\psi) \mathcal{F}(\|x\| \cos(\psi), \|x'\| \cos(\psi - \alpha)) d\psi$ (for $g = 1, \cos, \cos^2$, etc.), $Q \in SO(d)$ is such that $Qx = (0, \dots, \|x\|, 0)^T$, $Qx' = (0, \dots, \|x'\| \cos(\alpha), \|x'\| \sin(\alpha))^T$ and

$$\eta_{d,r} = \begin{cases} 2^{\lceil \frac{d-2}{2} \rceil} \pi^{\lfloor \frac{d-2}{2} \rfloor} & \text{if } r \text{ is even,} \\ 2^{\lfloor \frac{d-2}{2} \rfloor} \pi^{\lceil \frac{d-2}{2} \rceil} & \text{if } r \text{ is odd.} \end{cases}$$

Proof. If $Q \in SO(d)$ then by change of variables $a = Q^T \tilde{a}$ we have that

$$\begin{aligned}
\int_{a \in \mathbb{S}^{d-1}} aa^T \mathcal{F}(a^T x, a^T x') d\Omega &= \int_{\tilde{a} \in \mathbb{S}^{d-1}} Q^T \tilde{a} \tilde{a}^T Q \mathcal{F}(\tilde{a}^T Qx, \tilde{a}^T Qx') d\Omega \\
&= Q^T \left(\int_{\tilde{a} \in \mathbb{S}^{d-1}} \tilde{a} \tilde{a}^T \mathcal{F}(\tilde{a}^T \tilde{x}, \tilde{a}^T \tilde{x}') d\Omega \right) Q,
\end{aligned}$$

where $\tilde{x} = Qx$. Without loss of generality we thus assume that $x = (0, \dots, \|x\|, 0)^T$ and $x' = (0, \dots, \|x'\| \cos(\alpha), \|x'\| \sin(\alpha))^T$ where $\alpha = \arccos(x^T x' / \|x\| \|x'\|) \in [0, \pi]$. We now adopt hyperspherical coordinates $(\theta_1, \dots, \theta_{d-2}, \psi)$ where $\theta_i \in [0, \pi]$ and $\psi \in [0, 2\pi]$. The conversion between cartesian and spherical coordinates is given by:

$$\begin{aligned}
a_1 &= \cos(\theta_1) \\
a_2 &= \sin(\theta_1) \cos(\theta_2) \\
&\vdots \\
a_{d-1} &= \sin(\theta_1) \dots \sin(\theta_{d-2}) \cos(\psi) \\
a_d &= \sin(\theta_1) \dots \sin(\theta_{d-2}) \sin(\psi)
\end{aligned}$$

We also have that

$$\begin{aligned}
d\Omega &= \sin^{d-2}(\theta_1) \sin^{d-3}(\theta_2) \dots \sin(\theta_{d-2}) d\theta_1 d\theta_2 \dots d\psi \\
a \cdot x &= a_{d-2} \\
&= \|x\| \sin(\theta_1) \dots \sin(\theta_{d-2}) \cos(\psi) \\
a \cdot x' &= a_{d-2} \\
&= a_{d-2} \|x'\| \cos(\alpha) + a_{d-1} \|x'\| \sin(\alpha) \\
&= \|x'\| \sin(\theta_1) \dots \sin(\theta_{d-2}) (\cos(\psi) \cos(\alpha) + \sin(\psi) \sin(\alpha)) \\
&= \|x'\| \sin(\theta_1) \dots \sin(\theta_{d-2}) \cos(\psi - \alpha)
\end{aligned}$$

We now consider the integral $\int aa^T \mathcal{F}(a^T x, a^T x') d\Omega$. For any two indices $i \leq j \leq d-2$, we have

$$\begin{aligned}
&\int_{a \in \mathbb{S}^{d-1}} a_i a_j \mathcal{F}(a \cdot x, a \cdot x') d\Omega \\
&= \int_0^{2\pi} \dots \int_0^\pi \sin^{d-2+r}(\theta_1) \dots \sin^{1+r}(\theta_{d-2}) a_i a_j \mathcal{F}(\|x\| \cos \psi, \|x'\| \cos(\psi - \alpha)) d\psi d\theta_1, \dots, d\theta_{d-2} \\
&= \int_0^\pi \sin^{d-2+r+2}(\theta_1) d\theta_1 \dots \int_0^\pi \sin^{(d-1-i)+r+1}(\theta_i) \cos(\theta_i) d\theta_i \dots \int_0^\pi \sin^{(d-1-j)+r}(\theta_j) \cos(\theta_j) \theta_j \\
&\quad \dots \int_0^\pi \sin^{(d-1-(j+1))+r}(\theta_{j+1}) d\theta_{j+1} \dots \int_0^{2\pi} \mathcal{F}(\|x\| \cos \psi, \|x'\| \cos(\psi - \alpha)) d\psi.
\end{aligned} \tag{18}$$

This is now a product of $d - 1$ one-dimensional integrals. Since $\int_0^\pi \sin^s = 0$ if s is odd, we have that the integral (18) vanishes if $i \neq j$. If instead $i = j$, we use the fact that

$$\int_0^\pi \sin^s(t) \cos^2(t) dt = \frac{(s-1)!!}{(s+2)!!} \begin{cases} \pi & \text{if } s \text{ is even} \\ 2 & \text{if } s \text{ is odd} \end{cases}$$

and we deduce that

$$\begin{aligned} & \int_{a \in \mathbb{S}^{d-1}} a_i a_i \mathcal{F}(a \cdot x, a \cdot x') d\Omega \\ &= \eta_{d,r} \frac{(d-i+r-1)!! (d-i+r-2)!!}{(d+r)!!} \frac{r!!}{(d-i+r+1)!! (d-i+r-2)!!} \int_0^{2\pi} \mathcal{F}(\|x\| \cos \psi, \|x'\| \cos(\psi - \alpha)) d\psi. \\ &= \eta_{d,r} \frac{r!!}{(d+r)!!} \int_0^{2\pi} \mathcal{F}(\|x\| \cos \psi, \|x'\| \cos(\psi - \alpha)) d\psi. \end{aligned}$$

This proves the diagonal part in our expression for $\int a a^T \mathcal{F}(a^T x, a^T x') d\Omega$. All remaining terms as well as the two integrals $\int a \mathcal{F}(a^T x, a^T x') d\Omega$ and $\int \mathcal{F}(a^T x, a^T x') d\Omega$ follow from very similar (and slightly simpler) calculations. \square

We now apply Lemma 5 to compute the kernel K_∞ with the uniform initialization (8).

Proposition 6. *If $a \sim \mathcal{U}(\mathbb{S}^{d-1})$ and $b \sim \mathcal{U}([-k, k])$ and $\|x\|, \|x'\| < k$ then*

$$\begin{aligned} 2k \text{Vol}_{d-1}(\mathbb{S}^{d-1}) \cdot \mathbb{E}_{(a,b)} [ax + b]_+ [ax' + b]_+ &= \eta_{d,0} \frac{1}{(d-2)!!} \frac{2\pi}{3} k^3 + \eta_{d,2} \frac{2}{d!!} k \|x\| \|x'\| \pi \cos(\alpha) \\ &\quad + \eta_{d,3} \frac{3}{(d+1)!!} \left([E_1]_\tau^{\tau+\pi} + [E_2]_{\tau-\pi}^\tau \right) \\ 2k \text{Vol}_{d-1}(\mathbb{S}^{d-1}) \cdot \mathbb{E}_{(a,b)} [ax + b]_+ \mathbf{1}[ax' + b] a &= \eta_{d,1} \frac{2}{d!!} Q^T \begin{bmatrix} 0_{d-2} \\ k\pi \|x\| \\ 0 \end{bmatrix} + \eta_{d,2} \frac{3}{(d+1)!!} Q^T \begin{bmatrix} 0_{d-2} \\ G_1 \\ G_2 \end{bmatrix} \\ 2k \text{Vol}_{d-1}(\mathbb{S}^{d-1}) \cdot \mathbb{E}_{(a,b)} \mathbf{1}[ax + b] \mathbf{1}[ax' + b] a a^T &= \eta_{d,0} \frac{2\pi k}{d!!} I_d + \eta_{d,1} \frac{3}{(d+1)!!} Q^T \begin{bmatrix} \frac{1}{3} \delta I_{d-2} & 0 & 0 \\ 0 & \alpha & \beta \\ 0 & \beta & \gamma \end{bmatrix} Q \end{aligned} \tag{19}$$

where $\alpha = \arccos\left(\frac{x \cdot x'}{\|x\| \|x'\|}\right)$, $\tau = \arctan\left(\frac{\|x\| - \|x'\| \cos(\alpha)}{\|x'\| \sin(\alpha)}\right)$, $Q \in SO(d)$ is such that $Qx = (0, \dots, \|x\|, 0)^T$, $Qx' = (0, \dots, \|x'\| \cos(\alpha), \|x'\| \sin(\alpha))^T$ and

$$\eta_{d,r} = \begin{cases} 2^{\lceil \frac{d-2}{2} \rceil} \pi^{\lfloor \frac{d-2}{2} \rfloor} & \text{if } r \text{ is even,} \\ 2^{\lfloor \frac{d-2}{2} \rfloor} \pi^{\lceil \frac{d-2}{2} \rceil} & \text{if } r \text{ is odd.} \end{cases}$$

$$\begin{aligned}
E_1 &= \frac{1}{18} (\sin(\psi)^3 - 3 \sin(\psi)) \|x\|^3 + \frac{1}{24} \|x\|^2 \|x'\| (3 \sin(\psi + \alpha) + \sin(3\psi - \alpha) + 6 \sin(\psi - \alpha)) \\
E_2 &= \frac{1}{18} (\sin(\psi - \alpha)^3 - 3 \sin(\psi - \alpha)) \|x'\|^3 + \frac{1}{24} \|x\| \|x'\|^2 (\sin(3\psi - 2\alpha) + 3 \sin(\psi - 2\alpha) + 6 \sin(\psi)) \\
G_1 &= \frac{1}{3} \|x\|^2 \sin(\tau)^3 - \|x\|^2 \sin(\tau) + \frac{1}{2} \|x\| \|x'\| \sin(\alpha + \tau) + \frac{1}{6} \|x\| \|x'\| \sin(-\alpha + 3\tau) \\
&\quad + \|x\| \|x'\| \sin(-\alpha + \tau) - \frac{1}{12} \|x'\|^2 \sin(-2\alpha + 3\tau) - \frac{1}{4} \|x'\|^2 \sin(-2\alpha + \tau) - \frac{1}{2} \|x'\|^2 \sin(\tau) \\
G_2 &= \frac{1}{3} \|x\|^2 \cos(\tau)^3 - \frac{1}{2} \|x\| \|x'\| \cos(\alpha + \tau) - \frac{1}{6} \|x\| \|x'\| \cos(-\alpha + 3\tau) + \frac{1}{12} \|x'\|^2 \cos(-2\alpha + 3\tau) \\
&\quad - \frac{1}{4} \|x'\|^2 \cos(-2\alpha + \tau) + \frac{1}{2} \|x'\|^2 \cos(\tau) \\
\alpha &= \left[\|x\| \sin(\psi) - \frac{1}{3} \|x\| \sin(\psi)^3 \right]_{\tau}^{\tau+\pi} + \left[\frac{1}{4} \|x'\| \sin(\alpha + \psi) + \frac{1}{12} \|x'\| \sin(3\psi - \alpha) + \frac{1}{2} \|x'\| \sin(\psi - \alpha) \right]_{\tau-\pi}^{\tau} \\
\beta &= \left[-\frac{1}{3} \|x\| \cos(\psi)^3 \right]_{\tau}^{\tau+\pi} + \left[-\frac{1}{4} \|x'\| \cos(\alpha + \psi) - \frac{1}{12} \|x'\| \cos(3\psi - \alpha) \right]_{\tau-\pi}^{\tau} \\
\gamma &= \left[\frac{1}{3} \|x\| \sin(\psi)^3 \right]_{\tau}^{\tau+\pi} + \left[-\frac{1}{4} \|x'\| \sin(\alpha + \psi) - \frac{1}{12} \|x'\| \sin(3\psi - \alpha) + \frac{1}{2} \|x'\| \sin(\psi - \alpha) \right]_{\tau-\pi}^{\tau} \\
\delta &= \left[\|x\| \sin(\psi) \right]_{\tau}^{\tau+\pi} + \left[\|x'\| \sin(\psi - \alpha) \right]_{\tau-\pi}^{\tau}
\end{aligned} \tag{20}$$

Proof. The idea is to compute the integral with respect to the bias term b and then split the result into homogeneous expressions where Lemma 5 can be applied. In particular, assuming that $-k \leq s, t \leq k$:

$$\begin{aligned}
\mathcal{I}(s, t) &= \int_{-k}^k [s+b]_+ [t+b]_+ db = \int_{\min(s,t)}^k (s+b)(t+b) db \\
&= \underbrace{\left(\frac{1}{3} k^3 \right)}_{\mathcal{I}^0} + \underbrace{\left(\frac{1}{2} k^2 (s+t) \right)}_{\mathcal{I}^1} + \underbrace{\left(kst \right)}_{\mathcal{I}^2} + \underbrace{\left(\frac{1}{3} \min(s,t)^3 - \frac{1}{2} \min(s,t)^2 (s+t) + \min(s,t)st \right)}_{\mathcal{I}^3},
\end{aligned}$$

where we collected terms that have total degree 0, 1, 2, 3 in s, t . The computation of $\int_{b \in [-k, k]} \int_{\alpha \in \mathbb{S}^{d-1}} [ax+b]_+ [ax'+b]_+ db d\Omega = \int_{\alpha \in \mathbb{S}^{d-1}} I(a \cdot x, a \cdot x') d\Omega$ is now reduced to four one-dimensional integrals of the form $\int_0^{2\pi} \mathcal{I}^j(\|x\| \cos(\psi), \|x'\| \cos(\psi - \alpha)) d\psi$ where $\alpha = \arccos\left(\frac{x \cdot x'}{\|x\| \|x'\|}\right)$. The most tedious case is $j = 3$ where we need to compare $\|x\| \cos(\psi)$ and $\|x'\| \cos(\psi + \alpha)$. Assuming $0 \leq \alpha \leq \pi$, then $\|x\| \cos(\psi) < \|x'\| \cos(\alpha - \psi)$ holds if and only if $\tau \leq \psi \leq \tau + \pi$ where

$$\tau = \arctan\left(\frac{\|x\| - \|x'\| \cos \alpha}{\|x'\| \sin(\alpha)}\right).$$

From this we obtain that

$$\begin{aligned}
&\int_0^{2\pi} \mathcal{I}^3(\|x\| \cos(\psi), \|x'\| \cos(\alpha - \psi)) d\psi \\
&= \int_{\tau-\pi}^{\tau} \left(-\frac{1}{6} \|x'\|^3 \cos(\alpha - \psi)^3 + \frac{1}{2} \|x'\|^2 \|x\| \cos(\alpha - \psi)^2 \cos(\psi) \right) d\psi \\
&\quad + \int_{\tau}^{\tau+\pi} \left(-\frac{1}{6} \|x\|^3 \cos(\psi)^3 + \frac{1}{2} \|x\|^2 \|x'\| \cos(\psi)^2 \cos(\alpha - \psi) \right) d\psi
\end{aligned}$$

Expanding this integral we obtain the expressions for $\mathbb{E}_{(\alpha, b)} [ax+b]_+ [ax'+b]_+$ in the statement. The remaining integrals are

computed similarly by considering the homogeneous parts of

$$\begin{aligned}\mathcal{I}_t(s, t) &= \int_{-k}^k [s + b]_+ \mathbf{1}[s + b] db = \int_{\min(s, t)}^k (s + b) db \\ &= \frac{1}{2} k^2 + ks + \min(s, t)s - \frac{1}{2} \min(s, t)^2 \\ \mathcal{I}_{s, t}(s, t) &= \int_{-k}^k \mathbf{1}[s + b] \mathbf{1}[s + b] db = \int_{\min(s, t)}^k 1 db = k - \min(s, t).\end{aligned}$$

□

B.2. Gaussian Initialization

The Gaussian initialization (9) yields the following simpler formula for K_∞ . The first term is well known and is derived in [15]. The second and third terms are easily derived by taking derivatives of the first term.

Proposition 7. *If $a \sim \mathcal{N}(0, Id_{d-1})$ and $b \sim \mathcal{N}(0, 1)$, then*

$$\begin{aligned}2\pi \cdot \mathbb{E}_{(a, b)} [ax + b]_+ [ax' + b]_+ &= \|\tilde{x}\| \|\tilde{x}'\| (\sin(\tilde{\alpha}) + (\pi - \tilde{\alpha}) \cos(\tilde{\alpha})) \\ 2\pi \cdot \mathbb{E}_{(a, b)} [ax + b]_+ \mathbf{1}[ax' + b] a &= \|\tilde{x}\| (\sin(\tilde{\alpha}) + (\pi - \tilde{\alpha}) \cos(\tilde{\alpha})) \frac{x'}{\|\tilde{x}'\|} + (\pi - \tilde{\alpha}) \left(Id_d - \frac{x' x'^T}{\|\tilde{x}'\|^2} \right) x \\ 2\pi \cdot \mathbb{E}_{(a, b)} \mathbf{1}[ax + b] \mathbf{1}[ax' + b] aa^T &= (\pi - \tilde{\alpha}) Id + \sin(\tilde{\alpha}) \frac{x' x'^T}{\|\tilde{x}'\| \|\tilde{x}\|} \\ &\quad + \frac{1}{\sin(\tilde{\alpha})} \left(Id_d - \frac{x' x'^T}{\|\tilde{x}'\|^2} \right) \frac{xx^T}{\|\tilde{x}\| \|\tilde{x}'\|} \left(Id_d - \frac{xx^T}{\|\tilde{x}\|^2} \right),\end{aligned}\tag{21}$$

where $\tilde{x} = (x, 1)$, $\tilde{x}' = (x', 1)$, and $\tilde{\alpha} = \arccos\left(\frac{\tilde{x} \cdot \tilde{x}'}{\|\tilde{x}\| \|\tilde{x}'\|}\right)$.

C. Poisson Surface Reconstruction Kernel

In its simplest form, Poisson reconstruction of a surface [31], extracts the level set of a smoothed indicator function determined as the solution of

$$-\Delta f = \nabla \cdot V,$$

where V is a vector field obtained from normals n_i at samples x_i , and we use f to denote the (smoothed) indicator function as it plays the same role as f in (1). The equation above is closely related to (1): specifically, it is the equation for the minimizer of $\int_{\mathbb{R}^3} \|\nabla_x f(x) - V\|^2 dx$, i.e., the second term in (1), can be viewed as an approximation of this term by sampling at x_i . The screened form of Poisson reconstruction effectively adds the first term with $y_i = 0$, as the indicator function at points of interest is supposed to be zero. For the Poisson equation, the solution can be explicitly written as an integral

$$f(x) = \int_{\mathbb{R}^3} \frac{\nabla_z \cdot V(z) dz}{|x - z|}$$

The vector field V is obtained by interpolating the normals using a fixed-grid spline basis and barycentric coordinates of the sample points with respect to the grid cell containing it. This is equivalent to using a non-translation invariant non-symmetric locally-supported kernel $K_B(z, x)$:

$$V(z) = \sum_i K_B(z, x_i) n_i.$$

Let $B_{1,3}(x - c_j)$, $x \in \mathbb{R}^3$ be the trilinear basis $B_1(x^1 - c_j^1) B_1(x^2 - c_j^2) B_1(x^3 - c_j^3)$ function centered at a regular grid point c_j , and $B_{n,3}(x - c_j)$ be a tensor-product spline basis function of degree n defined in a similar way. (Note that in Lemma 3

in the main document, we slightly abuse notation, denoting B_1 as the trilinear basis and B_n as the degree- n spline basis). Poisson reconstruction uses $n = 1$ or $n = 2$ where

$$B_2(x) = \begin{cases} 0 & \text{if } x < -1.5 \\ \frac{1}{2}x^2 & \text{if } -1.5 \leq x < -0.5 \\ -\frac{1}{2} + x - (x-1)^2 & \text{if } -0.5 \leq x < 0.5 \\ \frac{5}{2} - x + \frac{1}{2}(x-2)^2 & \text{if } 0.5 \leq x < 1.5 \\ 0 & \text{if } 1.5 \leq z \end{cases} \quad (22)$$

Then $K_B(z, x) = \sum_j B_{1,3}(x - c_j)B_{n,3}(z - c_j)$. where only 8 terms corresponding to the vertices c_j of the grid cube containing x are nonzero. This yields the following expression for the kernel corresponding to Poisson reconstruction,

$$K_{\text{PR}}(x, x')_g = \int_{\mathbb{R}^3} \frac{\nabla_z K_B(z, x') dz}{|x - z|}$$

i.e., the convolution of the Laplacian kernel $1/|x - z|$ and the gradient of K_B . Using the identity $\nabla(f * g) = (\nabla f * g)$, we can write this as the gradient of $K_{\text{PR}}(x, x')_g$, defined as

$$K_{\text{PR}}(x, x') = \int_{\mathbb{R}^3} \frac{K_B(z, x') dz}{|x - z|} \quad (23)$$

To make it easier to understand the qualitative behavior of the kernel, replacing $K_B(z, x)$ with a radial kernel $B_n^1(|z - x|)$, with qualitatively similar behavior (see Figure 3) yields a translation-invariant radial approximation $K_{\text{PR}}^{\text{approx}}$ of the kernel K_{PR} , as the convolution of two radial kernels is a radial function.

$$K^{\text{approx}}(x, x') = \int_{\mathbb{R}^3} \frac{B_n^1(|z - x'|) dz}{|x - z|} \quad (24)$$

As both B_n^1 and the Laplace kernel are radial functions, their convolution is also radial. It can be expressed in a more explicit form using the relation between Fourier and Hankel transforms for radial functions. For $n = 3$, the Hankel transform is related to Fourier transform by [37]

$$s^{1/2} \mathcal{F}[g](s) = (2\pi)^{3/2} \mathcal{H}[g](s)$$

The Hankel transform is an involution, so the relationship for the inverse Fourier transform is similar. Writing $g * h = \mathcal{F}^{-1}[\mathcal{F}[g]\mathcal{F}[h]]$, we obtain the expression for the radial convolution in terms of one-dimensional integrals,

$$K^{\text{approx}}(x) = \mathcal{H}_s[s^{-3/2} \mathcal{H}_r[B_n^1]](|x|),$$

where we use $\mathcal{H}_r[1/r] = 1/s$. and $K_g^{\text{approx}}(x)$ is just the gradient of this, i.e., a derivative times $|x|/x$.

The RKHS norm for the space corresponding to this kernel is given by

$$\|f\|_{\mathcal{H}} = \int \frac{|\mathcal{F}[f]|^2}{\mathcal{F}[K^{\text{approx}}]} d\omega$$

with $\mathcal{F}[K^{\text{approx}}]$ obtained using the Hankel transforms as above.

D. RKHS Norm of the Neural Spline Kernel

We now discuss how $c(a, b)$ in (12) is related to the Laplacian of the function. If we make the mild assumption that our functions contain a linear and bias term (Lemma 8), then $c(a, b)$ is the Radon Transform of the laplacian of the function. Thus, the least norm minimizers of the least squares problem (1) are related to the laplacian of the function and the RKHS norm corresponds to the integral of the laplacian over hyperplanes in the domain. In our experiments, we added an option to include the linear and bias terms to the solution. They appear to have no effect on the final reconstruction. The derivation below is borrowed from [35].

Lemma 8. Let $f_{\text{lim}}(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ be an infinite-width, one hidden layer neural whose weights a, b are distributed according to the measure $c(a, b) : \mathbb{S}^{d-1} \times \mathbb{R} \rightarrow \mathbb{R}$

$$f_{\text{lim}}(x; c) = \int_{\mathbb{S}^{d-1} \times \mathbb{R}} [a^T x - b]_+ c(a, b) da db + v^T x + d. \quad (25)$$

Then, $f_{\text{lim}}(x)$ can always be rewritten as

$$\int_{\mathbb{S}^{d-1} \times \mathbb{R}} [a^T x - b]_+ c^+(a, b) da db + v'^T x + d' \quad (26)$$

where $c^+(a, b)$ is an even measure on $\mathbb{S}^{d-1} \times \mathbb{R}$, $v \in \mathbb{R}^d$, and $d \in \mathbb{R}$.

Proof. We can split the integral in f_{lim} into even and odd parts:

$$\begin{aligned} f_{\text{lim}}(x) &= \frac{1}{2} \int_{\mathbb{S}^{d-1} \times \mathbb{R}} ([a^T x - b]_+ + [a^T x - b]_-) c^+(a, b) da db \\ &\quad + \frac{1}{2} \int_{\mathbb{S}^{d-1} \times \mathbb{R}} ([a^T x - b]_+ - [a^T x - b]_-) c^-(a, b) da db \\ &\quad + v^T x + d \end{aligned}$$

where c^+ and c^- are the even and odd parts of c respectively. Observing that $[t]_+ + [-t]_+ = |t|$ and $[t]_+ - [-t]_+ = t$, we have that

$$f_{\text{lim}}(x) = \frac{1}{2} \int_{\mathbb{S}^{d-1} \times \mathbb{R}} (|a^T x - b| c^+(a, b) + (a^T x - b) c^-(a, b)) da db + v^T x + d \quad (27)$$

$$= \frac{1}{2} \int_{\mathbb{S}^{d-1} \times \mathbb{R}} |a^T x - b| c^+(a, b) da db + v'^T x + d' \quad (28)$$

$$= \int_{\mathbb{S}^{d-1} \times \mathbb{R}} [a^T x - b]_+ c^+(a, b) da db + v'^T x + d' \quad (29)$$

where $v' = v + \int_{\mathbb{S}^{d-1} \times \mathbb{R}} a c^-(a, b) da db$, $d' = d + \int_{\mathbb{S}^{d-1} \times \mathbb{R}} b c^-(a, b) da db$, and the last step holds because c^+ is even. \square

Using Lemma 8, we will consider without loss of generality, neural networks of the form (25) with even measures $c(a, b)$. We now give a few useful definitions and lemmas.

Definition 9. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$. The *Radon Transform* of f is

$$\mathcal{R}\{f\}(a, b) := \int_{a^T x = b} f(x) ds(x) \quad (30)$$

where $ds(x)$ is a measure on the $(d-1)$ -hyperplane $a^T x = b$. Intuitively the Radon transform represents a function in terms of its integrals along all possible hyperplanes.

Remark 10. Since the hyperplane $a^T x = b$ is the same as the hyperplane $-a^T x = -b$, the Radon transform is an even function. i.e. $\mathcal{R}\{f\}(a, b) = \mathcal{R}\{f\}(-a, -b)$.

Definition 11. Let $\varphi : \mathbb{S}^{d-1} \times \mathbb{R} \rightarrow \mathbb{R}$. The *Dual Radon Transform* of φ is the adjoint of the Radon Transform \mathcal{R}

$$\mathcal{R}^*\{\varphi\}(x) := \int_{\mathbb{S}^{d-1}} \varphi(a, x^T a) d\Omega \quad (31)$$

where $d\Omega$ is a measure on the $(d-1)$ -hypersphere \mathbb{S}^{d-1} . Intuitively the Dual Radon transform represents a function at x in terms of its integrals on all hyperplanes through x .

The Radon Transform satisfies the *intertwining property*. i.e. for any positive integer s

$$\mathcal{R}\{(-\Delta)^{\frac{s}{2}} f\} = (-\partial_b^2)^{\frac{s}{2}} \mathcal{R}\{f\} \quad (32)$$

Lemma 12. (From [41]) If $\varphi(a, b) = \varphi(-a, -b)$ is an even function mapping $\mathbb{S}^{d-1} \times \mathbb{R}$ to \mathbb{R} which is \mathcal{C}^∞ smooth and whose partial derivatives decrease at a rate faster than $\mathcal{O}(|b|^{-N})$ as $|b| \rightarrow \infty$ for any $N \geq 0$, then the Dual Radon Transform can be inverted using

$$\frac{1}{2(2\pi)^{d-1}} \mathcal{R}\{(-\Delta)^{\frac{d-1}{2}} \mathcal{R}^*\{\varphi\}\} = \varphi \quad (33)$$

Lemma 13. (From [35]) Let $f_{\text{lim}}(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ be an infinite-width, one hidden layer neural whose weights a, b are distributed according to the even measure $c(a, b)$. Then $c(a, b)$ can be expressed as

$$c(a, b) = \gamma_d \mathcal{R}\{(-\Delta)^{\frac{d+1}{2}} f_{\text{lim}}(x)\},$$

where $\mathcal{R}\{f\}(a, b)$ is the Radon Transform of f . In particular, for $d = 3$,

$$\gamma_d \mathcal{R}\{\Delta f_{\text{lim}}(x)\} = c(a, b)$$

Proof. The Laplacian of f_{lim} in is (25)

$$\Delta f_{\text{lim}}(x; c) = \int_{\mathbb{S}^{d-1} \times \mathbb{R}} \delta(a^T x = b) c(a, b) da db = \int_{\mathbb{S}^{d-1}} c(a, a^T x) d\Omega \quad (34)$$

which is precisely the Dual Radon Transform of $c(a, b)$. Since c is even, and assuming it decays rapidly with b , we can invert it using Lemma 12 yielding

$$c(a, b) = \gamma_d \mathcal{R}\{(-\Delta)^{\frac{d+1}{2}} f_{\text{lim}}(x)\}.$$

□

Corollary 14. The RKHS norm of the function f_{lim} is

$$\|f_{\text{lim}}\|_{\mathcal{H}} = \|c(a, b)\|_2 + \|v\|_2 + |d| = \left(\int_{\mathbb{S}^{d-1} \times [-k, k]} c(a, b)^2 d\Omega db \right)^{\frac{1}{2}} + \|v\|_2 + |d| \quad (35)$$