Surf-NeRF: Surface Regularised Neural Radiance Fields
Supplementary Materials

A. Deterministic Uniform Sphere Sampling

Sampling over spherical domains requires careful construction to avoid over- or under-sampling, particularly as the azimuth coordinate $\theta \to \pm\frac{\pi}{2}$. We experimented extensively, and found that traditional uniform sampling leads to unintended clusters of sampling directions which led to artefacts in the specular term. This can be seen in the random sampling row of Table 5 in the main paper. For this reason, we adapt the approach proposed by Frisch et al. [18] designed to sample a von Mises-Fisher (vMF) distribution in $\mathbb{S}^2$ to provide a uniform deterministic sample of a sphere.

The vMF distribution,

$$f(\mathbf{x}) = \frac{\kappa}{2\pi(e^{\kappa} - e^{-\kappa})} e^{\kappa \boldsymbol{\mu}^\top \mathbf{x}}, \tag{1}$$

is a probability distribution with mean $\boldsymbol{\mu} \in \mathbb{S}^2$ and concentration parameter $\kappa \geq 0$ on the 2-sphere. In the limit as $\kappa \to 0$, $f(\mathbf{x}) = (4\pi)^{-1}$, resulting in a uniform distribution.

Frisch et al. [18] sample this distribution using a Fibonacci-Kronecker lattice, with coordinates $\mathbf{x}_{FK} \in \mathbb{S}^2$,

$$\mathbf{x}_{F,i} = \begin{bmatrix} w \\ \sqrt{1-w^2}\cos\frac{2\pi i}{\Phi} \\ \sqrt{1-w^2}\sin\frac{2\pi i}{\Phi} \end{bmatrix}, \; i \in \{0, \ldots, N-1\} \tag{2}$$

where $\frac{1}{\Phi} = \frac{\sqrt{5}-1}{2}$ is the Golden Ratio, and

$$w = 1 + \frac{1}{\kappa}\mathrm{log1p}\left(\frac{2i-1}{2N}\mathrm{expm1}(-2\kappa)\right), \tag{3}$$

for $\mathrm{log1p}(x) = \log(1+x)$ and $\mathrm{expm1}(x) = \exp(x) - 1$. Since we want a uniform distribution, we take the limit $\kappa \to 0$, which results in,

$$\lim_{\kappa \to 0} w = w_{\kappa=0} = \frac{1 - 2i + N}{N}. \tag{4}$$

We wish to maintain the distribution of directions of the sphere but partition the samples through the volume of the unit ball. We introduce a discrete radius for each sample,

$$\mathbf{x}_i = \frac{1}{\log_2 N}(1 + i \bmod \log_2 N)\mathbf{x}_{F,i}, \tag{5}$$

to partition the samples into $\log_2 N$ shells in the unit ball.

We recognise that these shells, whilst uniform in direction, have decreasing density with increasing distance from the centre of the sphere. This helps in our use case for the sampling, with a higher density of samples closer to the centre of the ball (i.e. the surface). To avoid any bias that may arise in volumetric samples, we generate a rotation matrix which is uniformly distributed in $\mathrm{SO}(3)$, using [3]. Figure A.1 provides a visualisation of the direction and spatial coordinates for each sample using this approach.

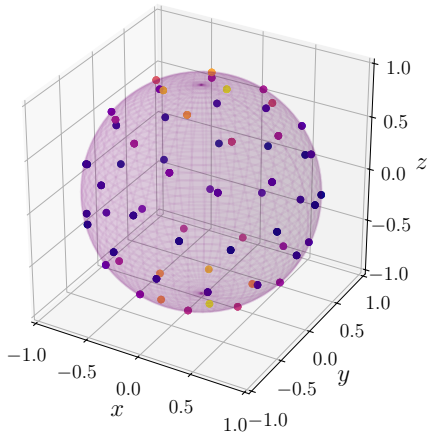
We initialise a sphere of N (in our case, $N = 32$) samples prior to the training loop, and at each iteration generate a unique rotation matrix for each ray which is applied to this sphere to provide the unique sampling. Rays that view the same part of the scene are therefore unlikely to sample to the same positions or directions during training.

B. Surface Sampling Behaviour as Samples Converge

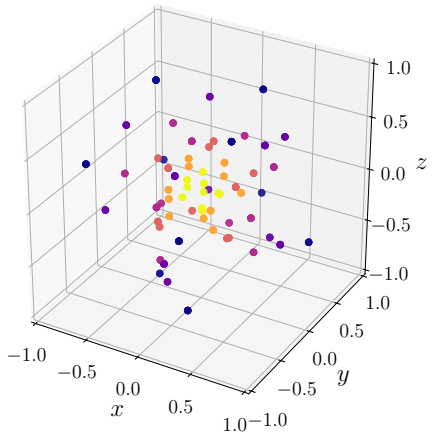
During training of a NeRF with integrated positional encoding [5, 6] (IPE), the frequencies of features increase as the ray samples converge to a final value and the modelled Gaussian regions of the space used for rendering reach minimum volumes. The proposed Surf-NeRF sampling scheme uses parameters from the integrated positional encoding to adapt the local region used for regularisation depending on the distribution of density along a ray. Localised distributions of density are regularised at finer scales, more strongly producing our intended effects later during training than earlier when the NeRF is still converging.

Where the sampling proposed for the fine NeRF model in MipNeRF360 [6] produces highly clustered values, the spacing between subsequent samples becomes small; that is $\Delta t \to 0$. The ideal surface representation within a NeRF is in fact a delta function along the ray, where density values are very large and highly clustered. Earlier during training, this distribution is more sparse. The evolution of the sample spacing is shown in Figure B.1.

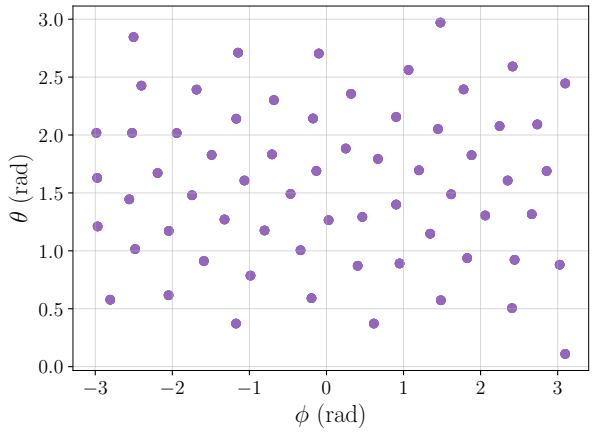
In this limit of convergence to a surface, the modelled Gaussians from IPE along a ray reach a minimum. Specifically, the sampling width t_δ between subsequent samples on the ray t_i and t_{i+1} approaches 0, as the mean sample



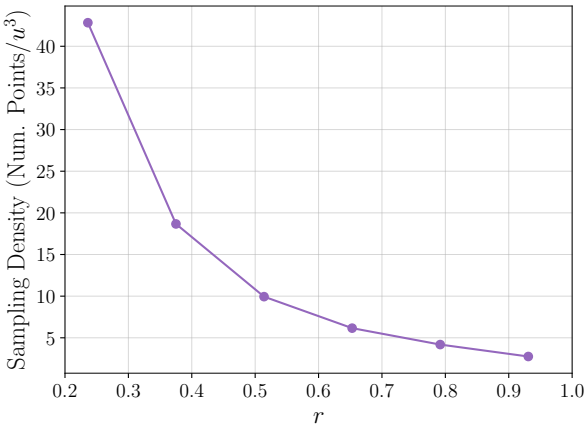
(a) 3D plot of deterministic samples on the sphere before applying the $\log_2 N$ radii partitions to sample the unit ball.



(b) 3D plot of deterministic sampling. These point locations are used in the proposed spatial sampling batch.



(c) Directions through which we sample. We see that the lattice uniformly distributes sample directions, taking into account the warping close to the poles.



(d) The density (number of points per unit volume) of each shell in the proposed spatial sampling batch. Samples closest to the surface have a higher density and therefore greater effect during regularisation.

Figure A.1. Example directional batch and its polar coordinate map, indicating samples used for regularisation and volumetric density of sampling.

approaches t_i,

$$t_\mu = \lim_{t_{i+1}\to t_i} \frac{t_{i+1}+t_i}{2} = t_i, \tag{6a}$$

$$t_\delta = \lim_{t_{i+1}\to t_i} \frac{t_{i+1}-t_i}{2} = 0. \tag{6b}$$

Evaluating the equations for mean and variance in MipNeRF [5], we obtain the following:

$$\lim_{t_{i+1}\to t_i} \mu_t = t_\mu, \tag{7a}$$

$$\lim_{t_{i+1}\to t_i} \sigma_t^2 = 0, \tag{7b}$$

$$\lim_{t_{i+1}\to t_i} \sigma_r^2 = \frac{\dot{r}^2 t_\mu^2}{4}, \tag{7c}$$

where $\dot{r}$ is the radius of the modelled view cone of a ray at the image plane. The samples turn from 3D anisotropic Gaussians to planar isotropic Gaussians in the scene orientated along the ray, thereby sampling in the radial direction only.

In Surf-NeRF, we use the radial standard deviation σ_r to scale our sampling sphere. As a result, our additional sampling scales to the certainty that a surface is present. When the samples are placed far apart as the NeRF is still converging to localised density along a ray, t_δ is non-zero and σ_r is greater than $\dot{r}t/2$. Regularisation therefore occurs over a

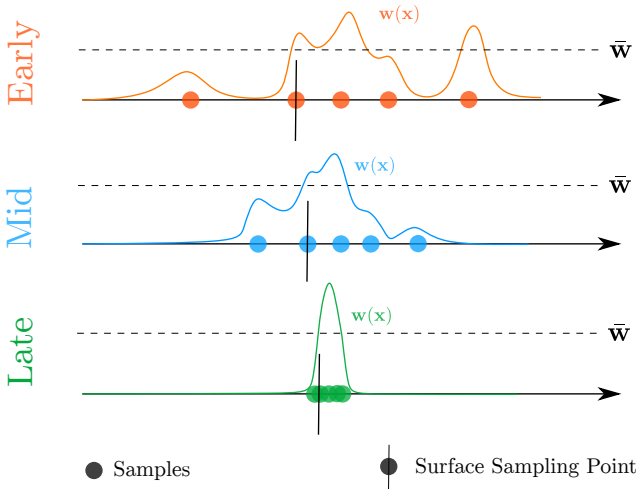


Figure B.1. The evolution of weights as the scene representation converges during training. Surf-NeRF uses a first surface assumption to determine the location of sampling, which is robust to outliers along the ray.

larger area, encouraging geometric smoothness and normal consistency to be based on samples further away, adding density but ablating finer details. As training progresses, σ_r approaches a minimum, ensuring sampling occurs within the bounds of a single pixel, thus retaining fine details and fine-tuning the representation.

C. Lambertian Bias

Positional encoding helps MLPs learn higher frequency information [37] and reflection encodings [35, 46, 47] encourage the separation of low-frequency colour information with view-dependent components. However, in practice, we find that the view-dependent term maintains a view-independent component across viewing angles. This is visualised in Figure C.1, and in Figure 5 in the main paper.

Early in training this phenomenon can be useful, allowing density to accumulate independently of diffuse colour and for surfaces with high roughness to develop low-frequency specular distributions. In many cases, these low frequency components remain in the view-dependent appearance of the scene given minimal photometric error and no loss regularising this component. This has the drawback, however, that the representation does not need to assign a constant view-independent colour to points in the scene allowing for ambiguous placement of scene geometry. For most use cases involving novel view synthesis, this is not a critical consideration.

However, by encouraging a unique Lambertian colour for each point in the scene where possible, we are able to better represent geometry by encouraging a view-independent appearance that must be consistent with all

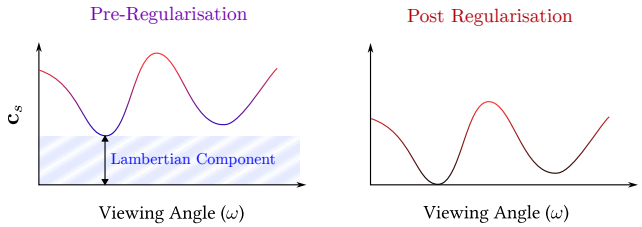


Figure C.1. Our motivation in regularising the view-dependent appearance for geometry within a NeRF. The specular term absorbs the Lambertian colour, increasing reliance on floaters and false geometry. Removing this Lambertian bias forces the model to place density in a more physically consistent manner, reducing reliance on floaters.

other observations of the scene. In essence, our regularisation minimises the magnitude of the specular term with the assumption that the view-dependent appearance should be sparse through viewing angles.

D. Total Variation of Unordered Samples on a Sphere

We apply total variation regularisation $\mathcal{L}_s$ on the specular colour $\mathbf{c}_s$ to produce more smoothly varying view-dependent appearances. Using the normal of the surface, we produce samples over directions that are in the hemisphere visible to cameras in the scene and formulate a total variation over this visible set of point. An example of this hemisphere is shown in Figure D.1.

To minimise total variation in specular colour, we select the k-nearest neighbours of directions on the hemisphere by angular distance. Selecting nearest neighbours produces the unique property that the regularisation is able to wrap around angles on the hemisphere. In essence, this is graph total variation over a subset of a hemispherical domain. Our implementation in JAX [14] allows gradients to propagate back to each sample, but not its neighbours thereby producing unique regularisation effects at each point.

We experimented with different numbers of k when formulating this regularisation term. We found minimal improvement in geometric quality and marginally reduced visual fidelity going to higher numbers of k above 3 most likely due to the small number of regularisation samples used and their comparative sparsity in the hemispherical domain.

E. Koala Dataset Details

Here we provide some additional details regarding our captured Koala dataset. This data set was captured using a Universal Robotics UR5e robotic arm, allowing ground-truth poses to be captured and the same trajectory to be used between different captured scenes up to repeatability accu-

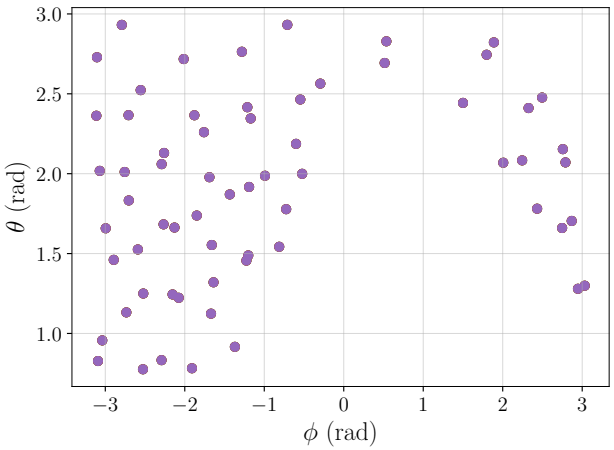


Figure D.1. Example directional polar coordinate map. The spliced forward and backward facing directions creates two interleaved sampling patterns, which is no longer regular. To address this, we use a graph total variation adapted for locations on the sphere accounting for irregularities.

racy of the robot arm (less than one tenth of a mm). Each scene contains approximately 40 training images and 10 test images. We specifically curate scenes to have increased shape-radiance ambiguity by using objects with more complex reflection, compared to the existing *Shiny Real* [46] dataset. This includes having large baselines between images, thereby only sparsely sampling the directional radiance across view angles and capturing objects at irregular distances, producing different resolutions at which specular effects are captured. The images were captured with a Basler acA1920-25uc camera, at constant gain and exposure in a controlled laboratory setting. This dataset provides a substantially more challenging benchmark given comparatively fewer images, more complex geometry, and more sparsity between poses to benchmark complex appearance reconstruction.

F. Details of Fine-Tuning Experiments

Since ZipNeRF has no Lambertian colour term, we do not include $\mathcal{L}_b$, and apply our sphere total variation regularisation $\mathcal{L}_s$ to the view-dependent colour $\mathbf{c}$, demonstrating applicability to models without reflection parameterisation. The spatial sampling terms depend only on the geometry and can be applied to other density-based neural fields without modification. We continue training at a fixed learning rate of 3.25×10^{-4} and present our results in Table 7.

G. Model Details

In Figure G.1, we provide a visual representation of our model structure incorporating the multiresolution hash encoding of PermutoSDF [42] over the cubic grid in ZipNeRF [7] and the physics-inspired structure of Ref-NeRF [46]. For our positionally encoded implementation, we use the same model parameters as those in the original Ref-NeRF paper.

From the base ZipNeRF implementation, we alter the hash encoding to use a permutohedral lattice hash grid. Our model maintains the two proposal and final NeRF networks at the same resolution as ZipNeRF. We retain the 512, 2048 and finally 8192 grid sizes for our lattice in that order. We also maintain the proposed sampling strategy of the NeRF from ZipNeRF (64 samples for each proposal network, and 32 for the final NeRF). Our experiments did not indicate substantial changes to the final quality of the NeRF with the same 128 final samples used in Ref-NeRF. In particular, in our first surface assumption for Surf-NeRF we note that the majority of the radiance along a ray should be the result of only a handful of samples clustered at the surface. Using a small number of final samples is in keeping with this assumption.

For the spatial MLP we use two layers with 128 neurons in each layer, a deeper network compared to base ZipNeRF helping the NeRF to learn the additional channels under the Ref-NeRF parameterisation. We use a 3-layer, 256 neuron wide MLP to learn the specular colour similar to ZipNeRF. The respective components from Ref-NeRF, namely the integrated directional encoding (IDE), tone mapping reflection calculation, and dot-product remain unchanged in their construction and addition to the network.

All weightings for losses in ZipNeRF [7] and Ref-NeRF [46] remain unchanged from previously published values. The Surf-NeRF regularisation terms have the following weightings λ: our density smoothness $\mathcal{L}_d$ and normal consistency $\mathcal{L}_n$ terms have weightings 10^{-1}, our specular bias term $\mathcal{L}_b$ has weighting $3 \cdot 10^{-2}$, and our specular total variation term $\mathcal{L}_s$ has weighting 10^{-3}.

Our hash-based implementations were trained for 25,000 iterations using a batch size of 2^{16}, adjusted for GPU memory using the linear scaling rule. Our positionally encoded implementations were trained for 250,000 iterations using a batch size of 2^{14} rays. Optimisation followed the same learning schedule as ZipNeRF [7] and MipNeRF360 [6] for the hash-based and positionally encoded based implementations, respectively.

H. Per-Scene Results

In this section, we present the per-scene results of our baseline and proposed methods across the *Shiny Objects*, *Shiny Real*, and *Koala* datasets. For the rendered *Shiny Objects* dataset we provide bold values demonstrating the improved geometric performance attained from Surf-NeRF in comparison to other baseline methods across both the traditional positional encoded variants and the hash-based methods.

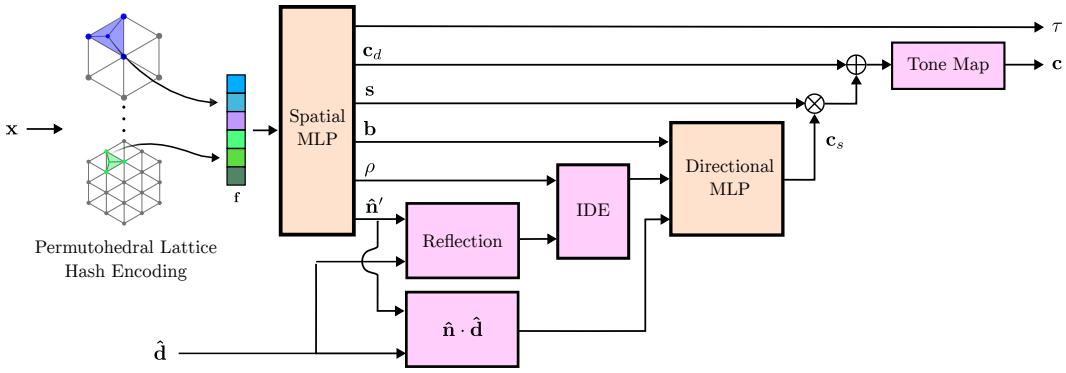


Figure G.1. Visual depiction of the adapted Surf-NeRF model structure. We replace the cubic multi-resolution hash encoding in ZipNeRF [7] with a multi-resolution permutohedral lattice structure, similar to PermutoSDF [42]. This is fused with the reflection encoding of Ref-NeRF [46]. We use the intermediate results for diffuse $\mathbf{c}_d$ and specular $\mathbf{c}_s$ colour terms to formulate the Surf-NeRF regularisation terms in terms of a Lambertian and specular scene appearance. Figure elements adapted from [42, 46].

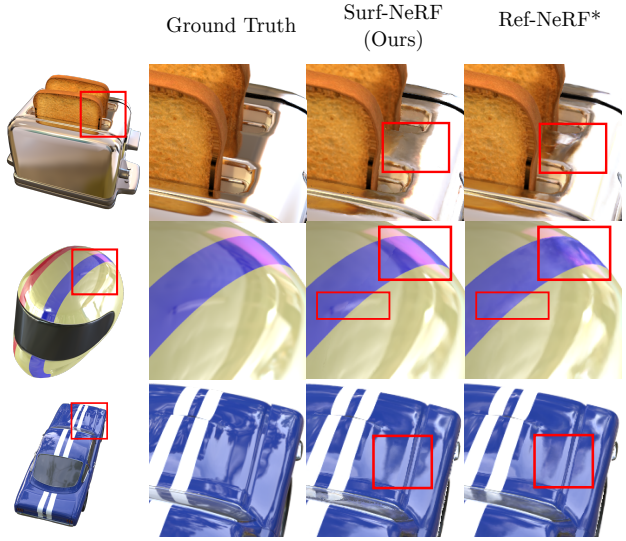


Figure H.1. Insets of renderings demonstrating improved fidelity of reflections compared to the Zip+Ref-NeRF baseline (denoted as Ref-NeRF*). Across all three scenes, Surf-NeRF is able to produce sharper, more realistic reflections as a result of substantially improving the underlying geometry in the scene.

For the real datasets, we demonstrate comparable visual performance to the existing methods with minimal degradation compared to the Ref-NeRF baselines we benchmark against.

In Figure H.1, we demonstrate the substantially improved sharpness and accuracy of reflections using our approach. Subtle reflections, for example, the light and window on the surface of the helmet and trees on the bonnet of the car, are substantially improved through producing reflections at the correct location of a surface in the scene, thereby producing realistic renderings.

Table H.1 outlines the results for the *Shiny Objects* [46] dataset. We show improved normals using our method across the positional encoded and grid-based approaches, and improved disparity RMSE scores. Figure H.2 provides additional qualitative results in this data set showing the separation of the Lambertian scene content and the qualitatively improved geometry across all data sets.

Figure H.3 compares to all grid-based baselines, showing improved geometry but remaining challenges with interreflection.

Figure H.4 shows the performance of positionally encoded models on low texture specular surfaces. Surf-NeRF yields drastically better surface geometry and normals.

Table 3 demonstrates comparable visual fidelity of our proposed regularisation compared to baseline approaches, with a small overall decrease due to the regularisation of the appearance and geometry in our curriculum learning framework.

Figure H.5 demonstrates qualitative performance in this data set, illustrating improved separation, depth, and comparable view fidelity.

Table H.3 provides quantitative results for our captured dataset, showing aspects of improved visual fidelity in difficult scenes. These are illustrated accordingly in Figure H.6.

In Figure H.7 we also show the impact of our curriculum ablation on the car scene, a comparatively difficult scene due to low texture and high reflection. We show that there is an optimal frequency to achieve improved normals and appearance separation, but higher frequencies more strongly separate this appearance at the detriment of geometry.

I. Additional Qualitative Benchmarks

In addition to the above results, we also present additional qualitative results across the MipNeRF360 [6] and DTU [1] datasets. In Figures I.1 and I.2, we demonstrate Surf-NeRFs

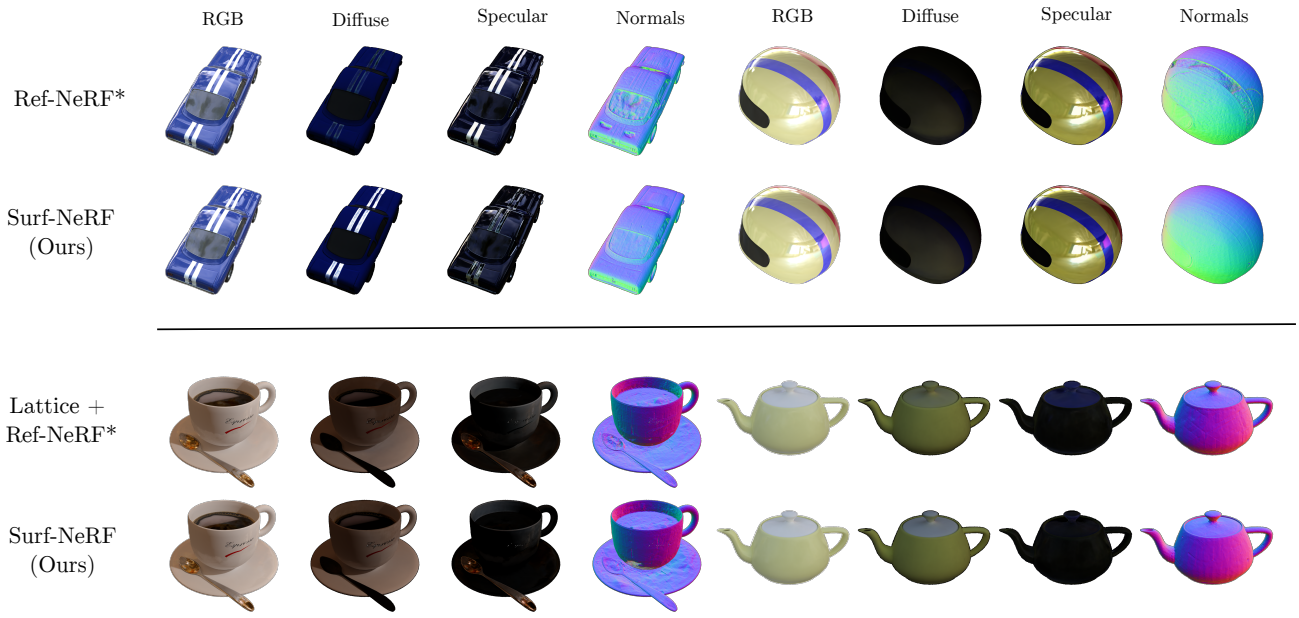


Figure H.2. Examples from the *Shiny Objects* dataset [46] for the hash based variants. Diffuse components like the white teacup surface and teapot lid are correctly removed from the specular term of the rendering, and normals show more complete surface structure. Ref-NeRF* indicates the Zip+Ref-NeRF baseline.

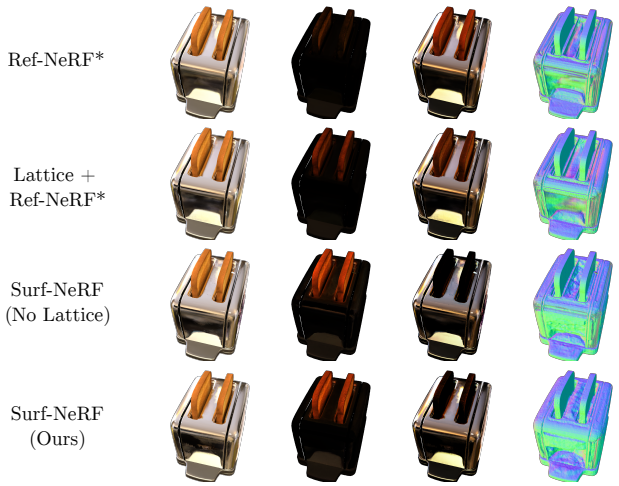


Figure H.3. Comparison across all reflection based baseline models and the proposed method on the Toaster scene. Applying the proposed regularisation dramatically improves the separation of diffuse and specular scene content, and improved surface normals over a baseline reflection parameterisation. Some elements with little view dependence like interreflection of close scene content remains a challenge to shape radiance ambiguity.

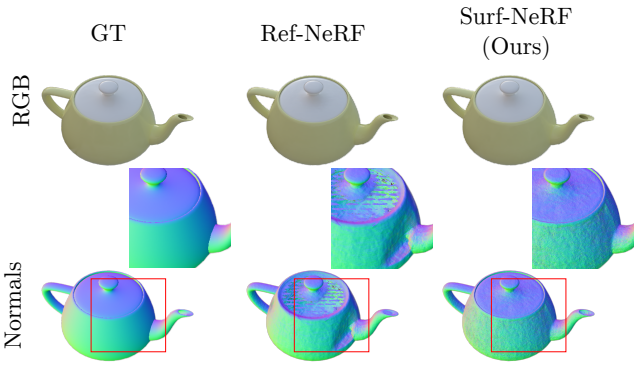


Figure H.4. Geometry results on the non-grid based models. The surface regularisation can resolve errors in geometry which stem from shape-radiance ambiguity e.g. in regions with low-frequency textures on the teapot lid, without compromising visual fidelity. This attains substantially improved normals in these regions.

ability to improve the underlying geometric representation of the scene, creating continuous surface normals and improved depth maps. We also demonstrate Surf-NeRFs ability to improve placement of intensity in the diffuse appearance, by removing view-dependent specularities from the bowl and moving intensity of the oven mits back to the diffuse channel.

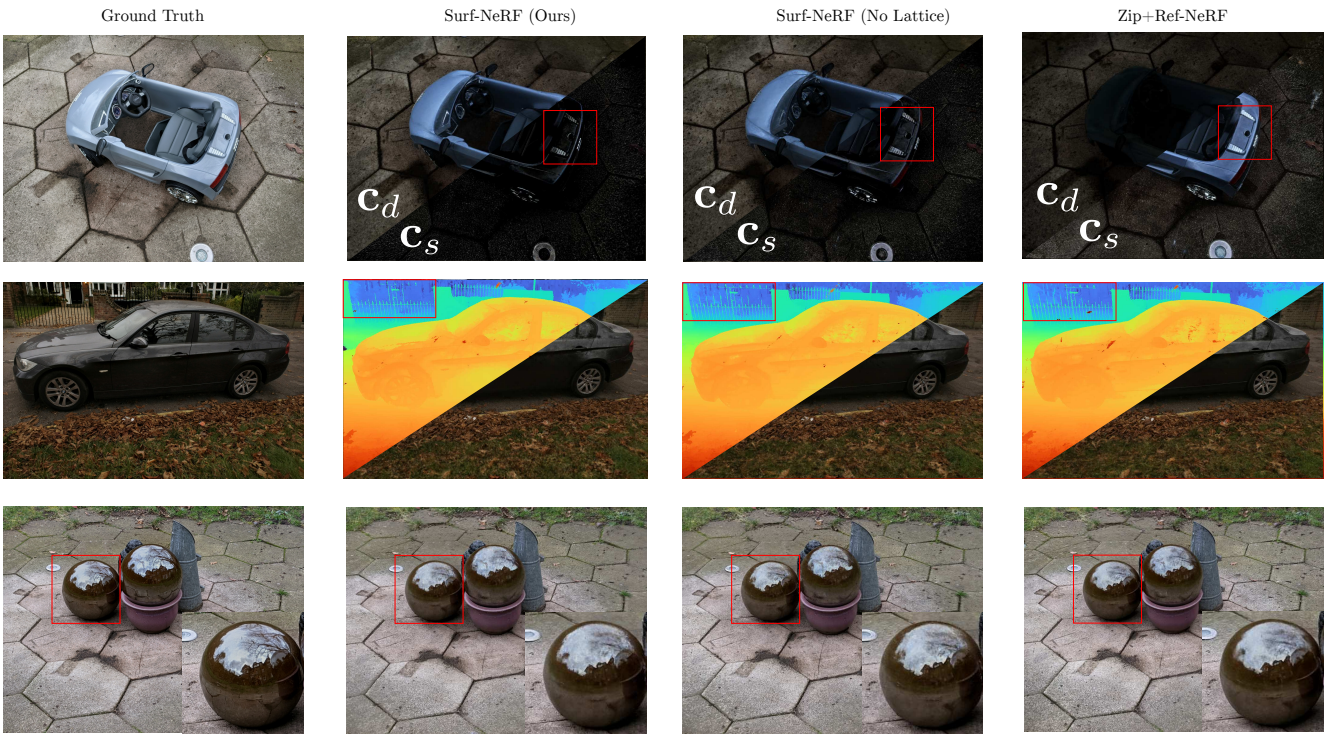


Figure H.5. Results on the *Shiny Real* dataset [46]. Surf-NeRF attains separation of diffuse and specular components $\mathbf{c}_d$ and $\mathbf{c}_s$ (top), a decreased reliance on floaters, more consistent depths, and qualitatively better representation of background content (middle), and comparable visual fidelity of reflective objects to other reflection parameterised NeRF variants.

Surf-NeRF (No Lattice)

Zip+Ref-NeRF

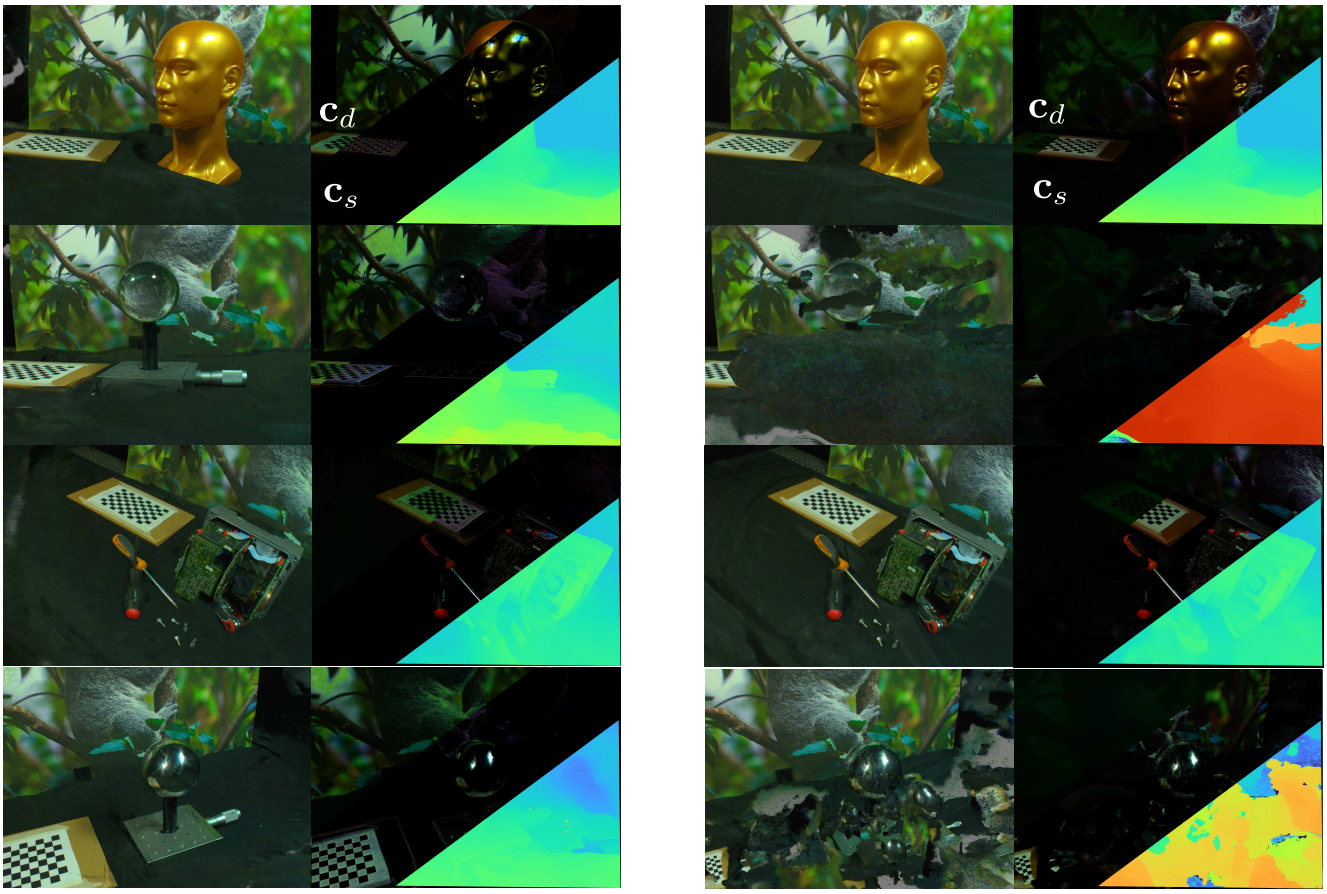


Figure H.6. Our captured *Koala* dataset in the no lattice case. We show improved separation of scene appearance between view-dependent and -independent content, more stable performance for scene content with high curvature and high view dependence. Renderings from models are on the left frame, with diffuse, specular and depth renderings on the right.

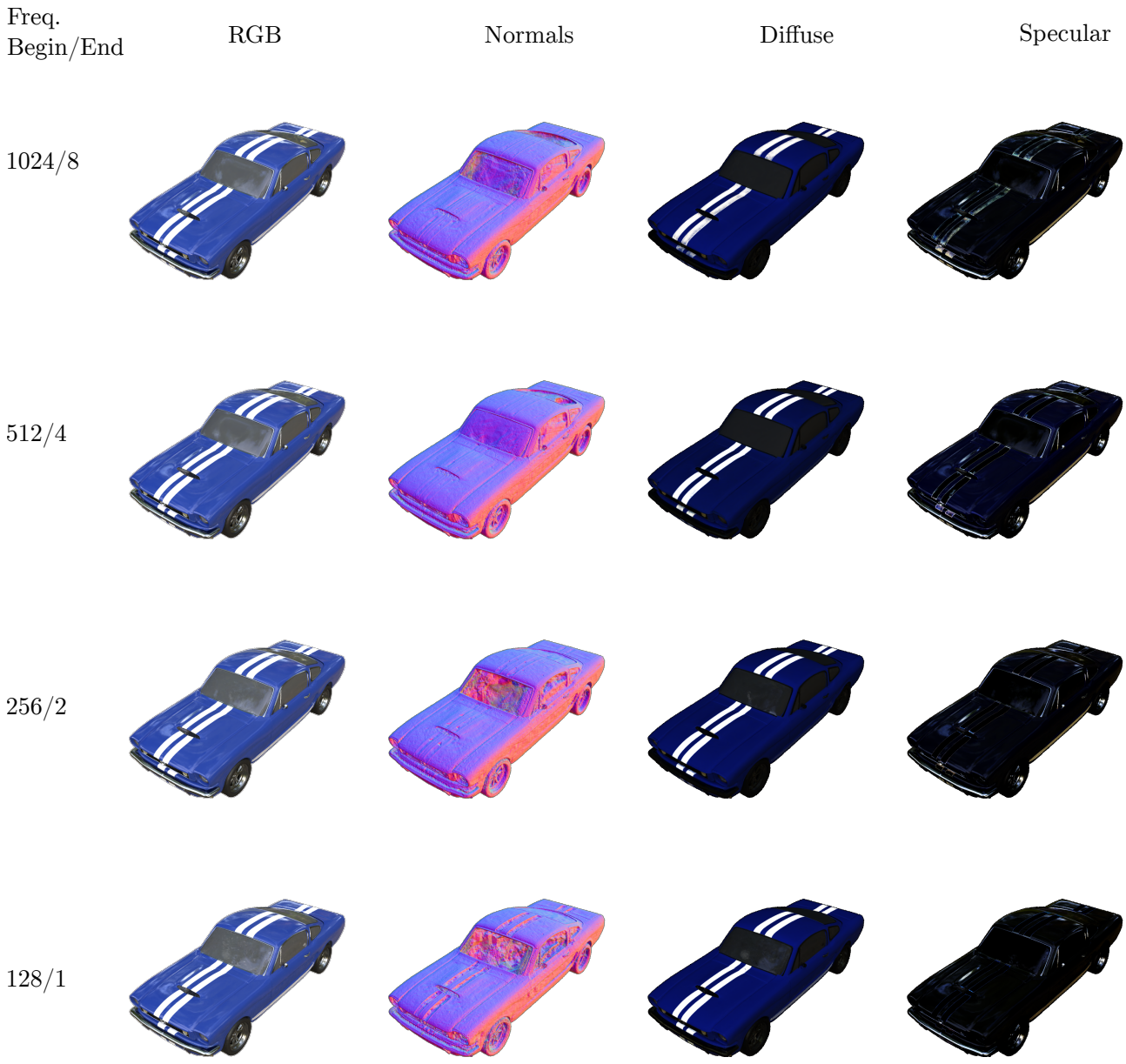


Figure H.7. Visualisation of the curriculum learning parameter study in the Table 6 on the car scene. As frequency increases, we see appearance separate more readily. We found that there were tradeoffs to the frequency of regularisation - as density is changed, the model must alter appearance to "take up" the changes to geometry. The 512/4 frequency was the pareto optimal state between strength of geometric improvement and the model's ability to alter appearance for the hash-based methods.

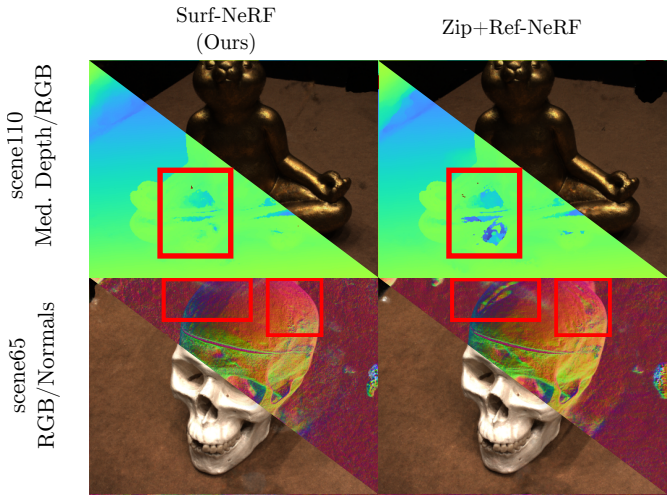


Figure I.1. Additional qualitative comparison between Surf-NeRF and the Zip+Ref-NeRF baseline on two scenes of the DTU dataset [1]. Surf-NeRF produces more accurate geometry in terms of depth and surface normals across both scenes by introducing more continuous surfaces.

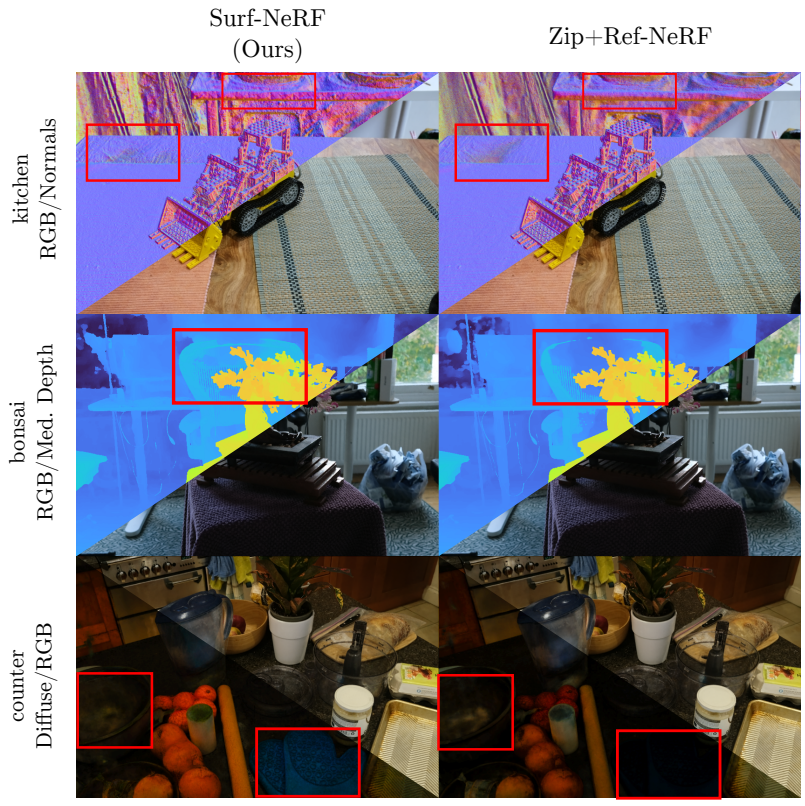


Figure I.2. Surf-NeRF exhibits similar performance to other 3D scenes, producing more defined and continuous surfaces as in the top and middle frames, and more accurately placing intensity of Lambertian scene content in the diffuse appearance of the scene.

Table H.1. Per-Scene Results on the *Shiny Objects* [46] dataset. Yellow, orange, red indicate third, second and first performing scores. Bold indicates best score for model type in disparity RMSE. We achieve improvements in all cases for disparity, overall improved normals and comparable visual fidelity results over the dataset.

	Model	PSNR	SSIM	MAE	RMSE
Car	MipNeRF	26.99	0.921	47.44	0.127
	MipNeRF+Diff	26.55	0.920	53.49	0.130
	Ref-NeRF	31.17	0.957	15.49	**0.123**
	Surf-NeRF (Ours)	30.80	0.955	14.99	**0.123**
	ZipNeRF	27.20	0.930	23.49	0.207
	Zip+Ref-NeRF	30.37	0.955	16.37	0.205
	Lattice+Ref-NeRF	30.30	0.957	15.16	0.209
	Surf-NeRF (No Lattice)	29.82	0.953	13.16	0.198
	Surf-NeRF (Ours)	29.85	0.955	11.08	**0.168**
Coffee	MipNeRF	31.36	0.966	31.10	0.124
	MipNeRF+Diff	31.45	0.966	38.97	0.120
	Ref-NeRF	33.79	0.973	13.50	**0.115**
	Surf-NeRF (Ours)	33.56	0.972	12.54	**0.115**
	ZipNeRF	30.79	0.977	13.48	0.207
	Zip+Ref-NeRF	32.56	0.971	11.95	0.203
	Lattice+Ref-NeRF	32.80	0.972	9.93	0.199
	Surf-NeRF (No Lattice)	31.99	0.968	14.79	**0.185**
	Surf-NeRF (Ours)	32.48	0.970	11.57	0.205
Helmet	MipNeRF	29.03	0.943	75.28	0.117
	MipNeRF+Diff	27.95	0.934	72.18	0.112
	Ref-NeRF	29.31	0.952	40.96	**0.109**
	Surf-NeRF (Ours)	30.09	0.958	26.68	**0.109**
	ZipNeRF	27.07	0.945	24.96	0.223
	Zip+Ref-NeRF	32.37	0.975	10.42	0.193
	Lattice+Ref-NeRF	32.31	0.976	10.22	0.218
	Surf-NeRF (No Lattice)	32.93	0.974	8.91	**0.192**
	Surf-NeRF (Ours)	35.61	0.983	4.49	0.200
Teapot	MipNeRF	45.90	0.996	67.07	0.152
	MipNeRF+Diff	45.62	0.996	69.10	0.145
	Ref-NeRF	46.24	0.997	15.82	**0.136**
	Surf-NeRF (Ours)	45.76	0.996	16.07	**0.136**
	ZipNeRF	46.11	0.997	9.10	0.192
	Zip+Ref-NeRF	46.09	0.997	7.62	**0.157**
	Lattice+Ref-NeRF	46.08	0.997	6.52	0.174
	Surf-NeRF (No Lattice)	44.23	0.996	6.17	**0.157**
	Surf-NeRF (Ours)	44.46	0.996	5.31	0.168
Toaster	MipNeRF	23.17	0.890	59.97	0.104
	MipNeRF+Diff	22.63	0.864	66.27	0.104
	Ref-NeRF	25.52	0.977	43.68	**0.102**
	Surf-NeRF (Ours)	24.76	0.911	41.20	**0.102**
	ZipNeRF	23.91	0.911	26.33	0.216
	Zip+Ref-NeRF	24.29	0.921	27.16	0.215
	Lattice+Ref-NeRF	24.17	0.922	27.06	0.209
	Surf-NeRF (No Lattice)	21.67	0.879	23.72	**0.198**
	Surf-NeRF (Ours)	20.76	0.869	20.55	0.202

Table H.2. *Shiny Real* [46] dataset results. Yellow, orange, red indicate third, second and first performing scores. Applying our regularisation does not significantly degrade visual fidelity compared to unregularised variants.

Scene	Model	PSNR	SSIM
Garden Spheres	MipNeRF	22.37	0.452
	MipNeRF+Diff	22.34	0.460
	Ref-NeRF	22.17	0.413
	Surf-NeRF (Ours)	22.33	0.443
	ZipNeRF	21.68	0.539
	Zip+Ref-NeRF	21.69	0.540
	Surf-NeRF (No Lattice)	20.51	0.486
	Surf-NeRF (Ours)	21.50	0.523
Sedan	MipNeRF	24.71	0.600
	MipNeRF+Diff	24.51	0.594
	Ref-NeRF	24.80	0.602
	Surf-NeRF (Ours)	24.72	0.614
	ZipNeRF	25.99	0.730
	Zip+Ref-NeRF	23.92	0.687
	Surf-NeRF (No Lattice)	23.71	0.684
	Surf-NeRF (Ours)	24.26	0.676
Toycar	MipNeRF	24.09	0.576
	MipNeRF+Diff	24.00	0.576
	Ref-NeRF	23.99	0.572
	Surf-NeRF (Ours)	24.06	0.571
	ZipNeRF	23.23	0.601
	Zip+Ref-NeRF	23.24	0.609
	Surf-NeRF (No Lattice)	23.20	0.620
	Surf-NeRF (Ours)	23.36	0.600

Table H.3. Surf-NeRF results on our captured *Koala* dataset. Bold indicates best performing scores. Regularisation helps the reflection-parameterised model around complex geometry and appearance on this dataset.

Scene	Model	PSNR	SSIM
Gold Head	ZipNeRF	**29.00**	0.680
	Zip+Ref-NeRF	28.34	0.681
	Surf-NeRF (No Lattice)	27.62	0.680
	Surf-NeRF (Ours)	26.61	**0.683**
Shiny Ball	ZipNeRF	**28.59**	0.641
	Zip+Ref-NeRF	15.47	0.435
	Surf-NeRF (No Lattice)	28.17	**0.646**
	Surf-NeRF (Ours)	27.78	0.643
Lidar Guts	ZipNeRF	27.42	0.664
	Zip+Ref-NeRF	**27.55**	0.665
	Surf-NeRF (No Lattice)	26.84	**0.666**
	Surf-NeRF (Ours)	25.88	0.642
Crystal Capture	ZipNeRF	26.14	**0.665**
	Zip+Ref-NeRF	23.40	0.577
	Surf-NeRF (No Lattice)	26.33	0.627
	Surf-NeRF (Ours)	**27.17**	0.648