

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

## Supplementary Materials for

# Hyper-NeRF: Hyperspectral Neural Radiance Fields with Continuous Radiance and Transparency Spectra

Anonymous ICCV submission

Paper ID 12454

## Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Implementation Details</b>	<b>1</b>
2.1. RGB Implementations . . . . .	2
<b>3. Training Details</b>	<b>2</b>
3.1. Commentary on the Tools Scene . . . . .	2
3.2. Loss Curves . . . . .	3
<b>4. Qualitative Example Results</b>	<b>3</b>

## 1. Introduction

In this work, we demonstrated that Neural Radiance Fields (NeRFs) can be naturally extended to hyperspectral data and are a well-suited tool for hyperspectral 3D reconstruction. The implementation details provided in this supplemental document describe our simple approach to hyperspectral NeRF, but we anticipate future works by the community will improve upon our baseline implementation using our to-be-published dataset, future larger datasets, additional architecture and hyperparameter tuning, and recent advances in NeRFs.

Our full code will be made publicly available for the camera ready version.

## 2. Implementation Details

We openly admit that significant improvements could be made on our implementation, re-emphasizing that our primary contribution is having demonstrated that NeRFs with continuous wavelength representations can work well on hyperspectral data.

We build upon nerfstudio's nerfacto implementation, from commit `ef9e00e`. The original nerfacto pipeline and field are shown in Figs. 1 and 2 respectively.

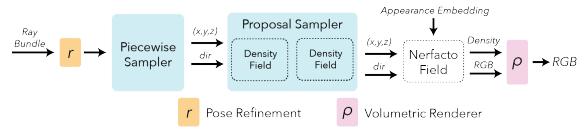


Figure 1. The original nerfacto pipeline (from [nerfstudio docs](#)) contains a proposal sampler, which is analogous to the “coarse” field from the original NeRF paper [2], and a “Nerfacto Field”, which is analogous to the primary network from the original NeRF paper ( $F_\Theta$ ).

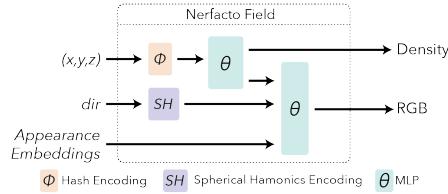


Figure 2. The original nerfacto field (from [nerfstudio docs](#)) is very similar to the original NeRF paper [2], but includes appearance embeddings [1] and uses slightly different encodings for the position and direction. This figure is reproduced in Fig. 2 of our main paper.

As briefly summarized in the main paper, we make minimal modifications to the pipeline and field. Using the notation from Section 5.4: Ablations,  $C_0$  is the stock nerfacto field;  $C_1$  only changes the rightmost MLP in Fig. 2 to output 128 channels in the last layer instead of 3;  $C_2$  changes the positional hash encoding ( $\phi$  in Fig. 2) to take 4 inputs instead of 3 (appending  $\lambda$ ) and changes the rightmost MLP to only have 1 output for  $c^\lambda$  instead of  $(r, g, b)$ ; and  $C$  is shown in Fig. 2 (bottom) of the main paper. For  $C$ , the sinusoidal encoding for  $\lambda$  is taken to have 8 terms (tested 2, 4, 8, 16 terms, with 8 performing marginally better than 4 and 16, and 2 significantly worse). Also for  $C$ , the component  $C(\lambda; \Theta_C)$  MLP from Fig. 2 of the main paper was taken to be identical to the rightmost MLP in Fig. 2 except

108 with the appropriate additional number of inputs to accommodate concatenating the sinusoidally encoded wavelength,  
109 and with only 1 output for  $c^\lambda$  instead of 3 for  $(r, g, b)$ . The  
110 latent vector  $\Theta_C$  was taken to be the same size as in the  
111 nerfacto implementation (15-dim), with increasing the size  
112 to 32 and 64 showing negligible performance improvement  
113 but increased training instability.  
114

115 Similarly,  $\sigma_0$  is the stock nerfacto field;  $\sigma_1$  only changes  
116 the left MLP in Fig. 2 to have 128 outputs;  $\sigma_2$  changes  
117 the positional hash encoding to take 4 inputs, and  $\sigma$  is as  
118 shown in Fig. 2 (bottom) of the main paper. The additional  
119 component  $\sigma(\lambda; \Theta_\sigma)$  MLP has 3 layers with 64-dim hidden  
120 layers and ReLU activations. The sinusoidally encoded  $\lambda$   
121 is shared with  $C$  and the latent  $\Theta_\sigma$  vector is shared with  
122 (identical to) the  $\Theta_C$  vector.  
123

124 Finally,  $P_0$  is the stock nerfacto proposal network while  
125  $P_\lambda$  augments the proposal network with the wavelength.  
126 For  $P_\lambda$ , the position is first run through a hash encoding  
127 and MLP as in  $P_0$ , except the MLP outputs a latent vector  
128 of dimension 7 instead of a scalar density. This latent  
129 vector is concatenated with a 2-term sinusoidally encoded  
130 wavelength and fed through a 2-layer network with 7-dim  
131 hidden layer to output a scalar density for inverse transform  
132 ray sampling. Like the original nerfacto pipeline, this sam-  
133 pling step occurs twice with identical architecture (but dif-  
134 ferent weights) proposal networks.  
135

136 Reiterating our implementation, our primary Hyper-  
137 NeRF implementation uses  $C(\lambda; \Theta_C)$ ,  $\sigma(\lambda; \Theta_\sigma)$ , and  $P_0$ ,  
138 which we find to produce good results while also enabling  
139 wavelength interpolation.  
140

## 2.1. RGB Implementations

141 **Erratum.** First, we apologize for the following error in  
142 the main paper that we will correct for the camera-ready  
143 version: For the caption in Table 1, we mistakenly state  
144 that our method outperforms the baseline for the more chal-  
145 lenge Tools and Origami scenes. We intended to portray  
146 that our approaches achieve very comparable performance  
147 to standard RGB nerfacto on all scenes, despite the fact that,  
148 for Ours-Cont and Ours-RGB, the wavelength bands are  
149 very far apart which should make learning *more* difficult.  
150 For Ours-Hyper, we achieve comparable performance on  
151 all scenes except Tools despite the fact that we are learning  
152 128 channels instead of just 3 while the number of learnable  
153 parameters is virtually identical to nerfacto and completely  
154 identical to Ours-Cont.  
155

156 **Pseudo-RGB wavelengths.** For the purposes of generating  
157 pseudo-RGB images, we use the wavelengths 622nm,  
158 555nm, and 503nm for R, G, and B channels respectively.  
159 Generating more accurate pseudo-RGB images by integrat-  
160 ing over the spectrum according to an image sensor sensi-  
161 tivity curve (as described in Section 5.5 of the main paper)  
162

163 would also be possible, but is unnecessary to demonstrate  
164 our results.  
165

166 **Hyper-NeRF RGB variation implementations.** For the  
167 purposes of making a quantitative comparison to standard  
168 RGB NeRF, Section 5.2 and Table 1 of the main paper  
169 present variations of our approach applied to just 3-channel  
170 (RGB) images instead of the full 128-channel hyperspectral  
171 data. As described in the caption of Table 1, “Ours-Cont”  
172 refers to our Hyper-NeRF implementation but trained on  
173 only 3 wavelengths, “Ours-RGB” refers to  $C_1, \sigma_1, P_0$  with  
174 3 output channels for both  $C_1$  and  $\sigma_1$ , and “Ours-Hyper”  
175 refers to our Hyper-NeRF implementation trained on all 128  
176 wavelengths. In the table for Ours-Hyper, PSNR and SSIM  
177 are evaluated over all 128 wavelengths while LPIPS is eval-  
178 uated only for the 3 channels closest to the red, green, and  
179 blue wavelengths according to our Pseudo-RGB procedure.  
180

## 3. Training Details

181 All networks were trained for 25000 steps, with 4096  
182 train rays per batch using the Adam optimizer. The pro-  
183 posal networks and field both used lr=1e-2, eps=1e-15, and  
184 weight decay=1e-6. Camera extrinsic and intrinsic opti-  
185 mization were both turned off, since evaluation metrics are  
186 skewed if camera parameters are modified. To accommo-  
187 date imperfect camera poses, after COLMAP, stock ner-  
188 facto was run on Pseudo-RGB images for 100000 steps with  
189 camera optimization turned on and the resulting camera  
190 pose corrections were saved and used in subsequent tests.  
191

192 Of the 48 images per image set, 43 were used for training  
193 and 5 withheld for evaluation. Each step, the 4096 training  
194 rays were sampled randomly from all 43 training images,  
195 except for row 5 of the ablations where the training rays  
196 were sampled from only 10 of the 43 training images each  
197 step, with the choice of 10 images being re-sampled every  
198 250 steps.  
199

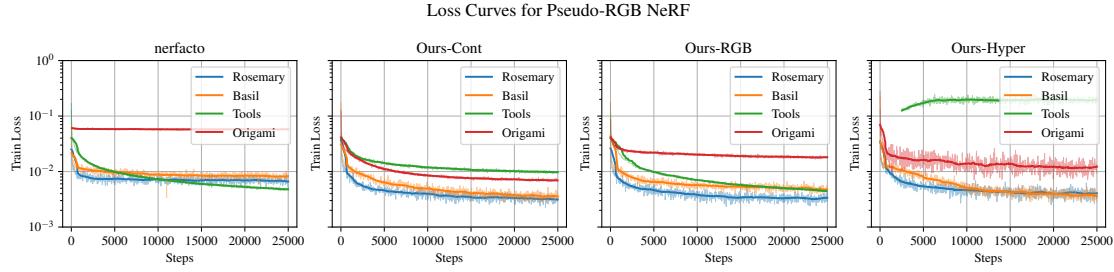
200 In some approaches, not all wavelengths could be run for  
201 every batch due to VRAM limits so a subset of wavelengths  
202 were sampled (randomly) for each batch, but every sampled  
203 wavelength was run for every ray in the batch. For rows 1  
204 and 2 of the ablations, every wavelength could be run ev-  
205 ery batch. For rows 3, 4 (Hyper-NeRF, ours), and 5, the  
206 number of wavelengths sampled per step were 8, 12, and 6,  
207 respectively.  
208

209 For evaluation, every wavelength of every pixel of the  
210 5 evaluation images were evaluated and compared for each  
211 scene.  
212

213 All tests were performed on an NVidia GeForce GTX  
214 3090, and most training runs took between 20min-60min.  
215

### 3.1. Commentary on the Tools Scene

216 The Tools scene experienced instabilities during training  
217 with several approaches including both Hyper-NeRF (ours)  
218

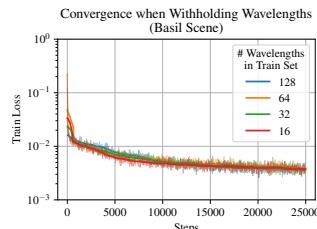
216  
217  
218  
219  
220  
221  
222  
223  
224

225 Figure 3. Loss curves for RGB NeRF correspond to the metrics from Table 1 in the main paper. Most scenes have converged by 25000  
226 steps except the Tools scene which appears to have difficulty converging for all methods except “Ours-Cont”, which is reflected in Table 1  
227 of the main paper.

228

229

230



231 Figure 4. Loss curves for Hyper-NeRF trained  
232 with a subset of wavelengths (analogous to Table  
233 2 in the main paper) shows that even training with  
234 only 1 out of every 8 wavelengths still has almost  
235 identical convergence rate w.r.t. number of steps.  
236

237

238

239

240

241

242

243

244 and nerfacto (RGB baseline). We anticipate that obtaining  
245 better camera intrinsics and extrinsics will correct this issue,  
246 since (a) every method had difficulty on this scene and (b)  
247 enabling camera pose optimization during NeRF training  
248 improved convergence for all methods. We plan to obtain  
249 better camera intrinsics by initializing COLMAP with the  
250 intrinsics obtained from other scenes, and we plan to ob-  
251 tain better camera extrinsics through a combination of tun-  
252 ing COLMAP parameters, utilizing turntable priors, and a  
253 longer NeRF-based camera pose refinement as described in  
254 3. The poor convergence on the Tools scene for all methods  
255 is illustrated in both Fig. 3 (green curves) and Fig. 5.

### 256 3.2. Loss Curves

257

258 To demonstrate that all methods were fairly trained until  
259 convergence, the loss curves corresponding to the met-  
260 rics given in the main paper are shown. As mentioned,  
261 the Tools scene appears to have difficulty converging for  
262 all methods including baseline nerfacto, suggesting possi-  
263 ble pre-processing (COLMAP) inaccuracy. This is evident  
264 both in the green curves of Fig. 3 and in the rightmost plot  
265 of Fig. 5.

266

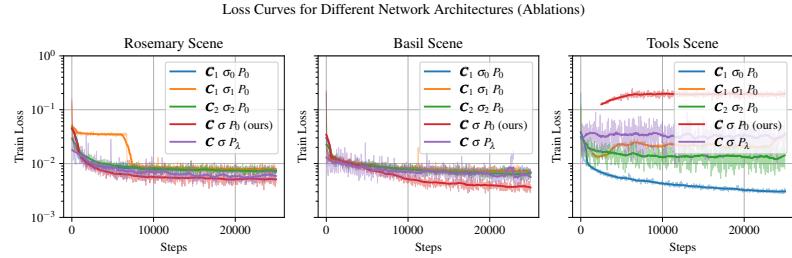
267

268

269

270 Exemplified by Fig. 5 (left, orange), one interesting ob-  
271 servation we found is that the  $C_1$  and  $\sigma_1$  architectures occa-  
272 sionally exhibit convergence followed by a second descent  
273 and convergence. Inspecting the evaluation images, we ob-

274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297



298 Figure 5. Loss curves for ablation testing (analogous to Table 3 in the main paper)  
299 shows that while the rosemary and basil scenes optimize well,  
300 the tools scene does not converge particularly well for any method, re-emphasizing the suspected  
301 pre-processing (COLMAP) inaccuracy.

302 serve the first convergence to be learning a scalar density  
303 field and the second convergence to be learning the color  
304 spectrum.

## 305 4. Qualitative Example Results

306 A selection of example images and videos are provided  
307 in both this pdf and in the enclosing zip folder to better  
308 gauge our results qualitatively. However, we emphasize  
309 again that our primary contribution is demonstrating that  
310 applying NeRF to hyperspectral data is a promising avenue  
311 for study. Therefore, we give 2 caveats: (1) visualizing  
312 pseudo-RGB results should not be compared to standard  
313 RGB NeRF results from other papers due to the many chal-  
314 lenges associated with hyperspectral cameras as described  
315 in Section 4.2 of the main paper, and (2) results should be  
316 interpreted as a starting point for future works to iterate  
317 upon rather than a comprehensive approach.

318 In Figs. 6 and 7, we can observe that Hyper-NeRF vis-  
319 ually appears “sharper” than all other approaches – the April-  
320 Tags clearly have much more detail in ours and the veins of  
321 the leaves also appear better resolved in Ours-RGB, Ours-  
322 Cont, and Ours-Hyper than the other approaches. These  
323 two figures depict pseudo-RGB representations of the hy-  
324 perspectral images rendered by our NeRFs in subfigures  
325 (e)-(l). Meanwhile, (b)-(d) depict the RGB renderings of 3-

324	channel, RGB NeRFs. Figs. 6 and 7 also evidence that Ab-	378
325	lation 5 (wavelength-dependent sample proposal network)	379
326	is never able to learn colors despite having already con-	380
327	verged (Fig. 5). Finally, Figs. 6 and 7 also evidence that the	381
328	hyperspectral approaches do not always have perfect color	382
329	accuracy, tinting the AprilTags slightly green, which sug-	383
330	gests that increasing the size of the color network or latent	384
331	vector may produce better results.	385
332	Note that Figs. 6 and 7 are on the next page.	386
333		387
334		388
335		389
336		390
337		391
338		392
339		393
340		394
341		395
342		396
343		397
344		398
345		399
346		400
347		401
348		402
349		403
350		404
351		405
352		406
353		407
354		408
355		409
356		410
357		411
358		412
359		413
360		414
361		415
362		416
363		417
364		418
365		419
366		420
367		421
368		422
369		423
370		424
371		425
372		426
373		427
374		428
375		429
376		430
377		431

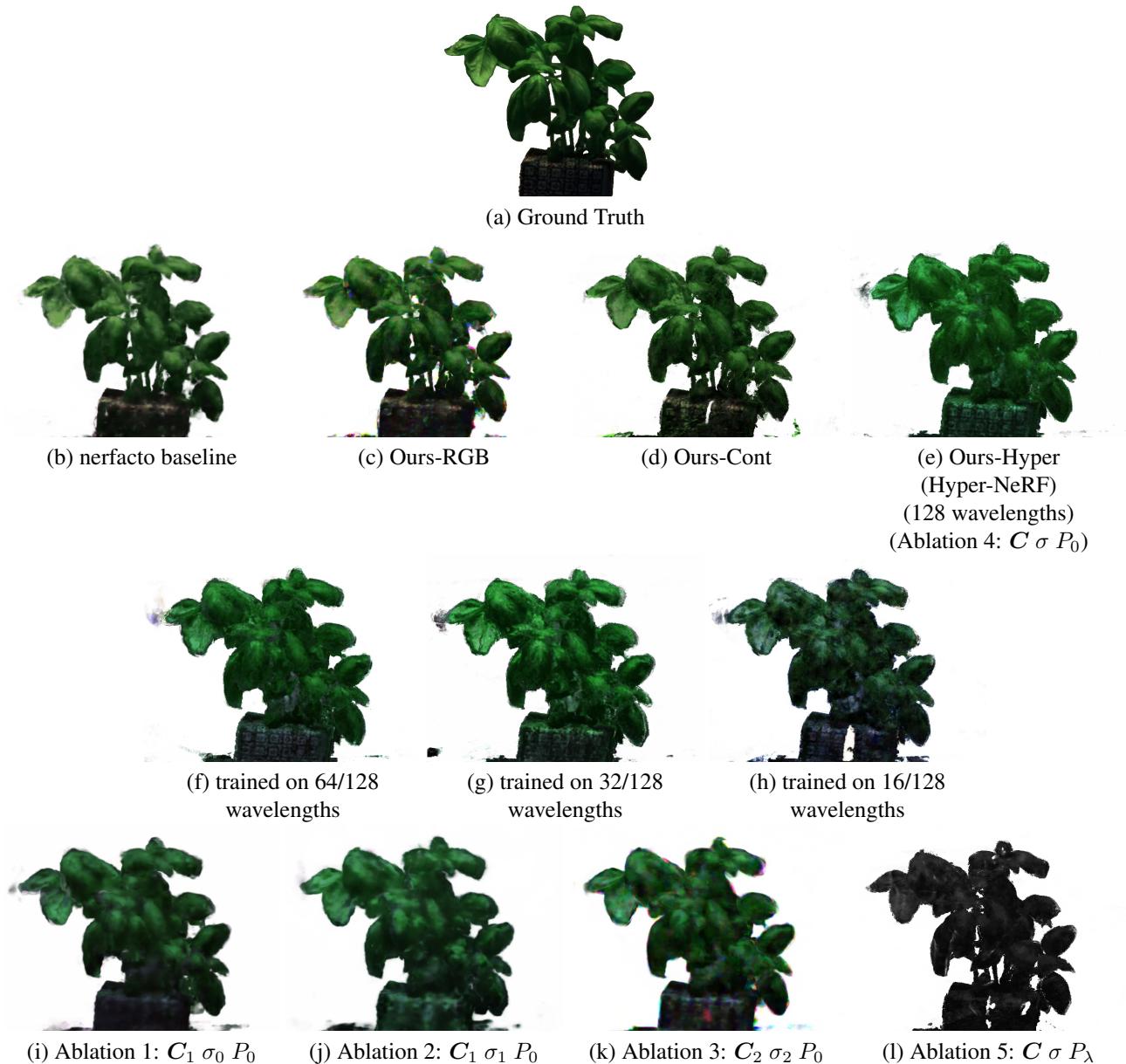
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511458  
459  
460  
461  
462  
463  
464  
465  
466  
467512  
513  
514  
515  
516  
517  
518  
519  
520468  
469  
470  
471  
472  
473  
474  
475  
476521  
522  
523  
524  
525  
526  
527  
528  
529477  
478  
479  
480  
481  
482  
483  
484  
485530  
531  
532  
533  
534  
535  
536  
537  
538  
539

Figure 6. Pseudo-RGB images of an evaluation image from the Basil scene for different methods demonstrates that our approach (e) is able to capture more detail in the leaves and AprilTags than all other approaches including the nerfacto baseline. (b)-(e) represent the 4 approaches from Table 1 of the main paper and Fig. 3. (e)-(h) represent the 4 approaches from Table 2 of the main paper and Fig. 4. (e) and (i)-(l) represent the 5 ablations from Table 3 of the main paper and Fig. 5. While the NeRF representations for (b)-(d) contain only RGB, all other subfigures are pseudo-RGB representations of the hyperspectral NeRFs.

540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593

594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647

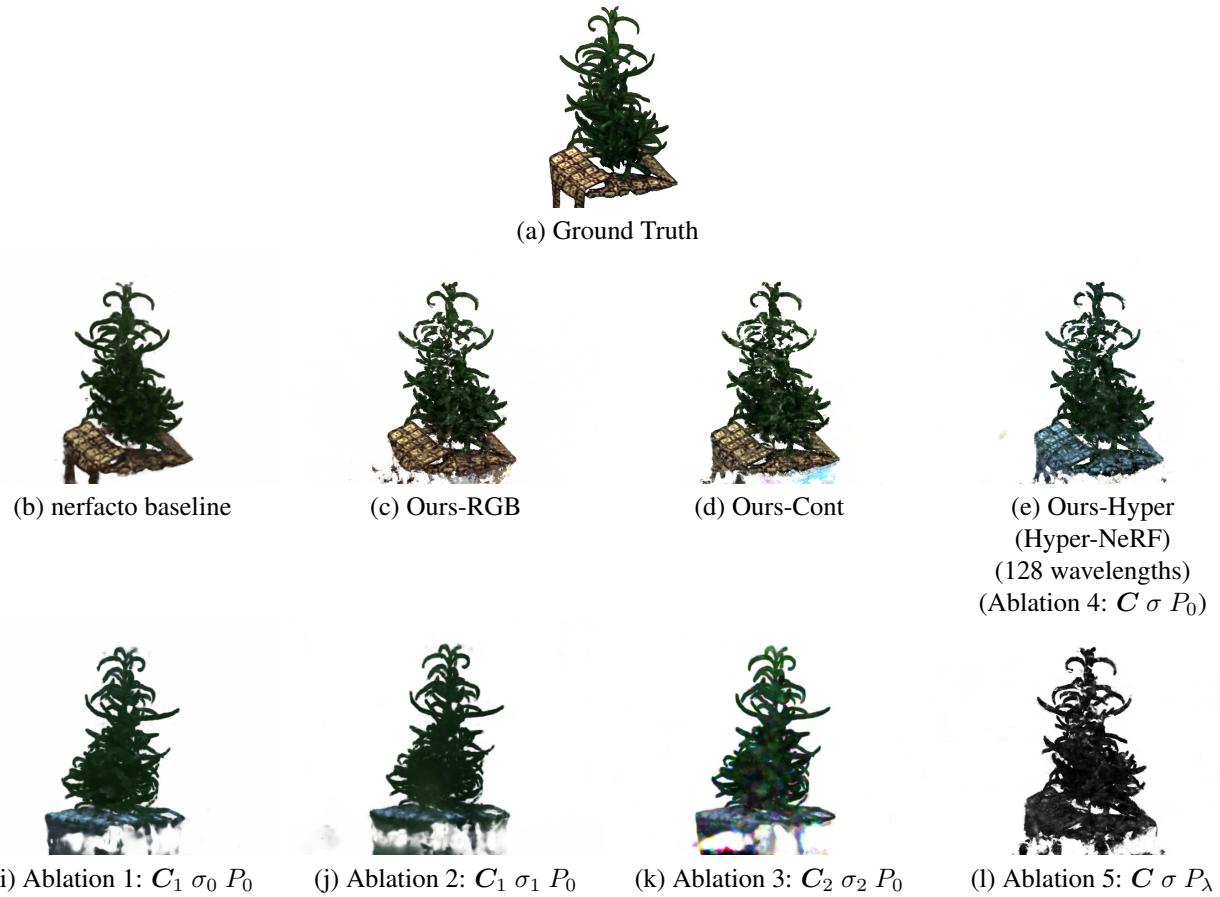


Figure 7. Same as Fig. 6 but for the Rosemary scene. Again, our approach appears to generate sharper results in the leaves and AprilTags than all other approaches including the nerfacto baseline. Contrary to the Basil scene, however, Ours-RGB and Ours-Cont also appear to generate results that have comparable sharpness to Hyper-NeRF.

648	<b>References</b>	702
649		703
650	[1] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi,	704
651	Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duck-	705
652	worth. NeRF in the Wild: Neural Radiance Fields for Uncon-	706
653	strained Photo Collections. In <i>CVPR</i> , 2021. 1	707
654	[2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik,	708
655	Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:	709
656	Representing scenes as neural radiance fields for view synthe-	710
657	sis. In <i>ECCV</i> , 2020. 1	711
658		712
659		713
660		714
661		715
662		716
663		717
664		718
665		719
666		720
667		721
668		722
669		723
670		724
671		725
672		726
673		727
674		728
675		729
676		730
677		731
678		732
679		733
680		734
681		735
682		736
683		737
684		738
685		739
686		740
687		741
688		742
689		743
690		744
691		745
692		746
693		747
694		748
695		749
696		750
697		751
698		752
699		753
700		754
701		755