# ViewFormer: NeRF-free Neural Rendering from Few Images Using Transformers

—

# Supplementary Material

Jonáš Kulhánek[1,2], Erik Derner[1], Torsten Sattler[1], and Robert Babuška[1,3]

[1] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
[2] Faculty of Electrical Engineering, Czech Technical University in Prague
[3] Cognitive Robotics, Faculty of 3mE, Delft University of Technology

https://jkulhanek.github.io/viewformer

In this supplementary material, we give more details on the results presented in the main paper and provide more details on the network architecture. First, in Sec. A, we present additional qualitative results on various datasets. We also show examples of context views used to render the final view. The attached video is described in Sec. B. We include the camera pose estimation results on the 7-Scenes dataset [12] in Sec. C, and we also show qualitative results of the novel view synthesis task on the same dataset. In Sec. D, we present an ablation study. We also show how the performance increases with larger context sizes. In Sections E and F, we include additional results on the ShapeNet dataset and the Shepard-Metzler-Parts-7 (SM7) dataset, respectively. Quantitative results of the codebook model are given in Sec. G. Finally, we give details on the training hyperparameters and architecture of the models in Sections H and I.

## A Qualitative results

We add qualitative results to the ones presented in the paper (see Fig. 1, 6, and 8 in the main paper). We show the context views together with the rendered images on the InteriorNet [16], the Common Objects in 3D (CO3D) [23], and the 7-Scenes [12] datasets. The generated images are displayed in Fig. 1, Fig. 2, and Fig. 4, respectively. We also show images generated with full context sizes in Fig. 3. It is important to note that all the visualizations, including the video, were rendered on previously unseen scenes (objects).

The images rendered on the largest and most complex dataset – InteriorNet, although slightly blurry, resemble the ground truth (GT) images well. For the 7-Scenes dataset, the trained model overfitted the data, and the quality of the generated images was not as good as on other datasets. Notice how the image rendered on CO3D is smoother than the ground truth image. In the case of the flower pot (Fig. 2), we can see that the model could not represent the particular shape and used a simpler shape instead. This is an intriguing property of the

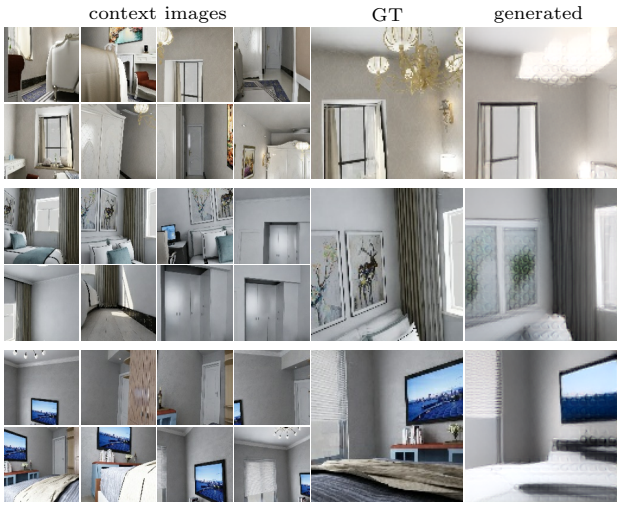


**Fig. 1.** Visualization of the model trained on the InteriorNet dataset [16]. We show the images generated with context size 8 while the model was trained with context size 19

model which in the case of incomplete information uses its large prior to achieve more realistic renderings at the cost of being less similar to the real object.

## B Attached video

We attach a video file[4] showing the generated images on various datasets. The video contains the results generated on the ShapeNet, CO3D, InteriorNet, and 7-Scenes datasets. On the ShapeNet dataset, we compare our model with PixelNeRF [35]. We render video sequences of rotating objects using the same three context views. For the CO3D dataset, we show video sequences of rotating objects using 9 context views. We also show how the model changes its prediction given more context views. Unfortunately, we cannot compare with PixelNeRF [35] because the method was not able to converge properly on the dataset (see Sec. 4 in the main paper). Also, we cannot compare with NerFormer [23] because the source code is not publicly available. Finally, we show the results on the InteriorNet dataset as well as on all scenes from the 7-Scenes dataset.

One might expect that with the discrete codebook codes the learned representation would be quantized and an arbitrary pose could not be represented by the model. However, from the sequences generated on the ShapeNet dataset, we can see that this problem does not occur and the model is able to capture the motion, smoothly transitioning between the true poses. Therefore, although the codes are discrete, they can represent a continuous range of objects' orientations and positions. It is interesting to see that our approach is occasionally

[4] https://jkulhanek.com/viewformer/video.html

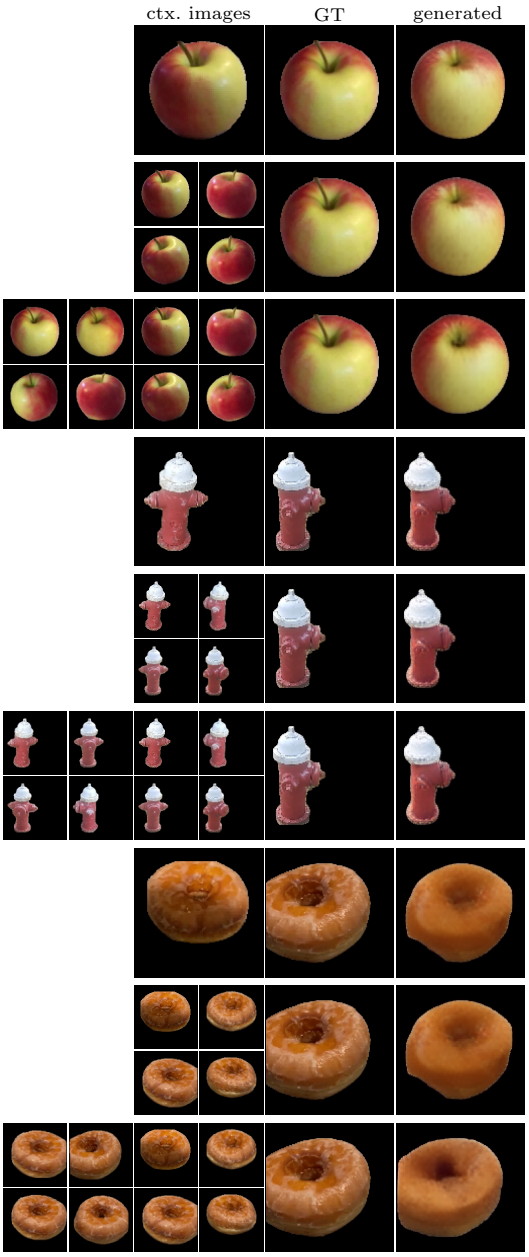


**Fig. 2.** Visualization of the model trained on the CO3D dataset [23]. We show the images generated with context sizes 1, 4, and 8 while the model was trained with context size 9

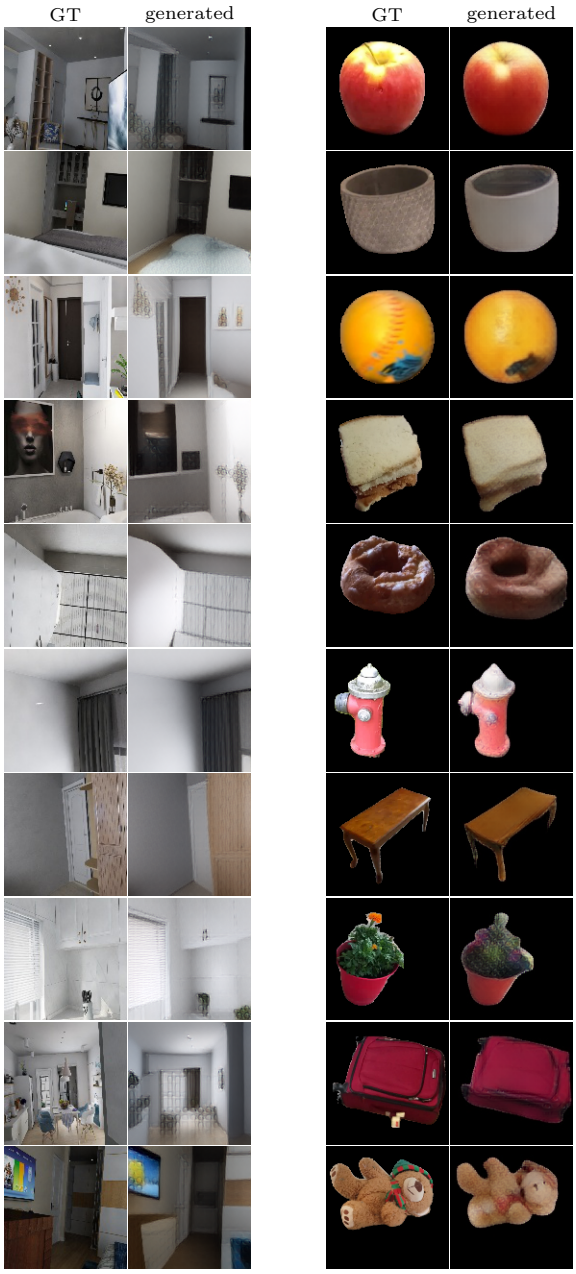


**Fig. 3.** Images generated on the InteriorNet dataset (**left**) with context size 19 and the CO3D dataset (**right**) with context size 9. For the CO3D evaluation, we used the model trained on all categories

**Table 1.** Camera pose estimation accuracy on the 7-Scenes dataset [12], reported as the mean median position (in meters) and orientation (in degrees) errors over all scenes. We report results with an InteriorNet pre-trained codebook ('-in') and a codebook fine-tuned on 7-Scenes ('-7s'). We further compare a simple decoding scheme (random context views) with a variant that uses the top-10 most similar training images for each query view ('top10'), identified via image retrieval

| Method | All Pos/Ori | Chess Pos/Ori | Fire Pos/Ori | Heads Pos/Ori | Office Pos/Ori | Pumpkin Pos/Ori | Kitchen Pos/Ori | Stairs Pos/Ori |
|---|---|---|---|---|---|---|---|---|
| ViewFormer-in | 0.24/10.49 | 0.16/8.03 | 0.24/11.35 | 0.17/13.23 | 0.25/10.33 | 0.23/8.20 | 0.31/11.01 | 0.30/11.28 |
| ViewFormer-in-top10 | 0.19/7.82 | 0.13/6.36 | 0.22/10.27 | 0.17/10.85 | 0.17/6.42 | 0.19/6.26 | 0.21/6.62 | 0.21/7.97 |
| ViewFormer-7s | 0.23/8.46 | 0.15/6.31 | 0.23/10.03 | 0.19/12.68 | 0.23/7.69 | 0.19/5.59 | 0.27/7.75 | 0.31/9.18 |
| ViewFormer-7s-top10 | 0.17/6.68 | 0.12/4.85 | 0.20/8.65 | 0.17/10.41 | 0.15/5.11 | 0.16/4.78 | 0.18/5.01 | 0.22/7.93 |
| Oracle-top10 | 0.21/10.01 | 0.18/9.16 | 0.27/10.37 | 0.12/11.44 | 0.22/8.33 | 0.24/8.20 | 0.26/9.72 | 0.19/12.85 |
| PoseNet [14] | 0.44/10.4 | 0.32/8.12 | 0.47/14.4 | 0.29/12.0 | 0.48/7.68 | 0.47/8.42 | 0.59/8.64 | 0.47/13.8 |
| MapNet [4] | 0.18/6.56 | 0.09/3.24 | 0.20/9.29 | 0.12/8.45 | 0.19/5.45 | 0.19/3.96 | 0.20/4.94 | 0.27/10.57 |
| LENS [18] | 0.05/2.5 | 0.04/2.0 | 0.03/1.5 | 0.02/1.5 | 0.09/3.6 | 0.08/3.1 | 0.07/3.4 | 0.03/2.2 |
| MS-Transformer [27] | 0.18/7.28 | 0.11/4.66 | 0.24/9.6 | 0.14/12.19 | 0.17/5.66 | 0.18/4.44 | 0.17/5.94 | 0.26/8.45 |
| RelocNet [1] | 0.21/6.72 | 0.12/4.14 | 0.26/10.4 | 0.14/10.5 | 0.18/5.32 | 0.26/4.17 | 0.23/5.0 | 0.28/7.53 |
| CamNet [8] | 0.04/1.69 | 0.04/1.73 | 0.03/1.74 | 0.05/1.98 | 0.04/1.62 | 0.04/1.64 | 0.04/1.63 | 0.04/1.51 |
| DenseVLAD [26,31] | 0.26/13.1 | 0.21/12.5 | 0.33/13.8 | 0.15/14.9 | 0.28/11.2 | 0.31/11.3 | 0.30/12.3 | 0.25/15.8 |
| DenseVLAD+Int.[26] | 0.24/11.7 | 0.18/10.0 | 0.33/12.4 | 0.14/14.3 | 0.25/10.1 | 0.26/9.42 | 0.27/11.1 | 0.24/14.7 |
| DSAC* [2] | 0.03/1.36 | 0.02/1.10 | 0.02/1.24 | 0.01/1.82 | 0.03/1.15 | 0.04/1.34 | 0.04/1.68 | 0.03/1.16 |
| hloc [24] | 0.03/1.09 | 0.02/0.85 | 0.02/0.94 | 0.01/0.75 | 0.03/0.92 | 0.05/1.30 | 0.04/1.40 | 0.05/1.47 |
| Active Search [25] | 0.04/1.18 | 0.03/0.87 | 0.02/1.01 | 0.01/0.82 | 0.04/1.15 | 0.07/1.69 | 0.05/1.72 | 0.04/1.01 |

not color consistent from frame to frame, *e.g.*, see the police car at time 0:07. We believe that the cause of this problem may stem from the codebook. It was trained using a perceptual loss, which might be less sensitive to colors [9]. On the InteriorNet dataset (time 3:02), look at the pictures on the wall. The model first generates a window in place of the pictures, and with more context views, it replaces the window with two pictures. This illustrates well how the model improves its prediction given more context views.

## C 7-Scenes evaluation

In order to evaluate the performance of our approach on the task of camera pose estimation, we present the results on a localization benchmark dataset – 7-Scenes [12] (*cf.* Sec. 4 in the main paper). We trained two models – one with a fine-tuned codebook and the other one with the InteriorNet-trained codebook. For all models, we used context size 19. We have evaluated the method on all views from the test set of each of the 7 scenes and used the views from the training set as context images. Generated images can be seen in Fig. 4.

For localization, we have experimented with different strategies for obtaining the context view required by our approach: by default, we simply randomly select 19 training images as context for each test image. We further evaluate a variant that uses the top-10 most similar images identified via image retrieval with DenseVLAD [31] descriptors (indicated as "-top10"). The remaining 9 context images are randomly selected from the training images. We also experimented with using the top-19 retrieved images but found this approach to work worse. We attribute this to the fact that the images of the 7-Scenes datasets are taken

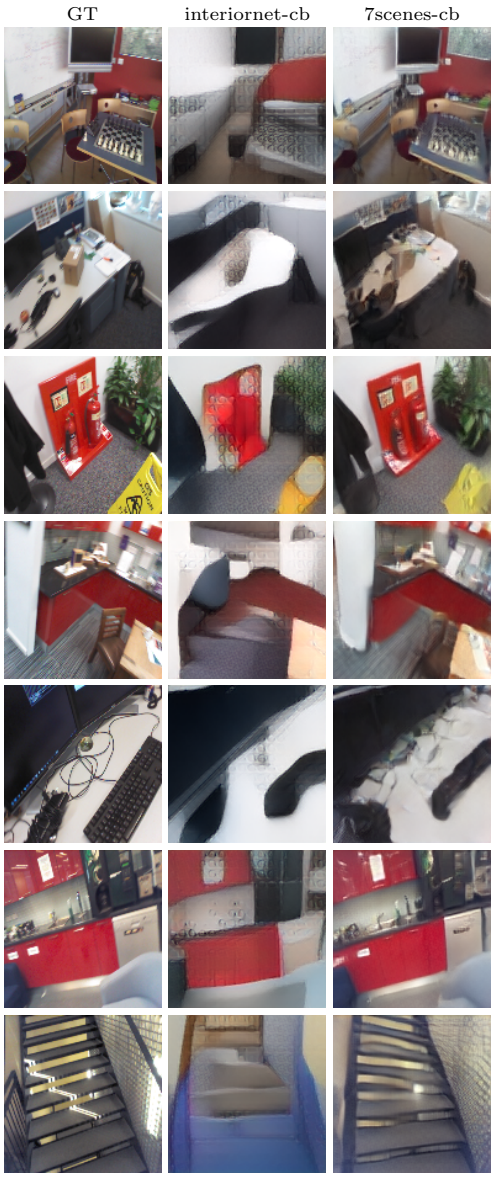


**Fig. 4.** Evaluation of the transformer model on the 7-Scenes dataset [12]. We display the ground-truth image (**GT**), the image generated using a codebook trained only on the InteriorNet dataset (**interiornet-cb**) and the image generated by a model with codebook fine-tuned on the 7-Scenes (**7scenes-cb**). For the visualization the context size was set to 19

in sequences and that there is little viewpoint variation between the top-19 retrieved images.

We evaluate variants where the codebook is trained only on InteriorNet (indicated as "-in") and where the codebook is fine-tuned on the training images of 7-Scenes ("-7s"). As can be seen in Tab. 1, using a fine-tuned codebook improves performance. Similarly, using the top-10 retrieved images leads to more accurate camera poses. For evaluation, we follow the common practice and report the median position and orientation error per scene, as well as the mean median position and mean median orientation error over all the scenes.

To better understand the performance of our approach, we compare it against an oracle. Given the top-10 retrieved images via DenseVLAD, the oracle selects the retrieved image with the smallest position and the smallest orientation error. As shown in Tab. 1, our approach outperforms the oracle on most scenes. This implies that the model is able to interpolate the context views such that it generates a pose that is closer to the query than any other in the context.

Tab. 1 also includes comparison with various baselines. Absolute pose regression techniques [4,14,18,27] train a CNN to directly regress the camera pose for a given input image. Our approach performs similarly well or better than these baselines, with the exception of LENS [18], which uses additional training data in the form of images rendered from novel viewpoints. Our approach also typically outperforms the two image retrieval-based baselines (DenseVLAD and DenseVLAD + Int.) They were proposed in [26] as a form of sanity check for absolute pose regression approaches.

Similar to our approach, relative pose regression approaches [1,8] estimate the pose of the test image *wrt.* a set of context views. These context views are obtained by finding the most similar training images using image retrieval. Our approach performs similarly well (and often better) as RelocNet [1], which also uses a single forward pass to regress relative poses (between pairs of images). CamNet [8] uses a more complicated pipeline consisting of coarse and fine relative pose regression stages, which results in higher accuracy.

Structure-based approaches use 2D-3D matches between pixels in a test image and 3D scene points [3,24,25]. These approaches currently represent the state-of-the-art in terms of pose accuracy and are more accurate than pose regression-based techniques. In contrast to the other baselines, they store the 3D structure of the scene. Overall, the results show that our approach achieves a similar level of pose accuracy as comparable methods.

## D Ablation study

We compare our model with alternative architectures to validate the design choices we made. We also demonstrate how the quality of predictions improves with larger context sizes. The InteriorNet dataset [16] was used for all evaluations because of its large size. The context size was 19.

**Different model variants.** We compare variants of our approach trained for only one of the two tasks – image generation and localization – on the Interi-

orNet dataset [16]. We also evaluate the importance of the proposed branching attention by training alternative language models (LMs) that do not use it. As discussed in Sec. 1 in the main paper, one way to train the transformer without the branching attention is to have a purely autoregressive (causal) LM [20,32]. These models were successfully applied to similar tasks [11,19,22]. We also train another alternative – masked LMs – that benefits from the same inference speed as our method [7]. In particular, the following models are compared:

- **ViewFormer** – our approach with both localization and image generation enabled.
- **ViewFormer no-loc** – our approach without localization.
- **ViewFormer no-imagen** – our approach without image generation.
- **Causal LM** – the same transformer model with autoregressive decoding. Instead of decoding all tokens at once, we model the probability distribution over the next image token given all previous tokens [20,32].
- **Causal LM + masked loc.** – causal LM with added localization. For the localization, we mask the poses of three random views from the training batch and attach a regression head to the last token of each image.
- **Masked LM** – the same transformer model with masked decoding (without the branching attention). We randomly mask three views from the training sequence and train the model to recover it. Note that the model is optimized for a single context size (previous variants optimized for all context sizes).
- **Masked LM + masked loc.** – masked LM with added localization. For the localization, we mask the poses of three random views from the training batch and attach a regression head to all image tokens. The resulting poses are averaged in the same way as in ViewFormer.

The results (averaged over all test scenes) are shown in Tab. 2. We also include a qualitative comparison in Fig. 5. As can be seen, training without the localization task improves image quality, whereas there is little difference in terms of pose accuracy between training with or without the generation.

Our method outperforms both causal LM and masked LM in image generation performance and localization accuracy. Note that our decoding is much faster compared to causal LM because we decode all tokens at once (see Section 1 in the main paper). For a causal LM, generating a single view takes 10 s even when using cache. Compare this to 93 ms for the ViewFormer. Compared to masked LM, our model has the same inference speed, but the added benefit of being optimized for all context sizes. Masked LM can be optimized for one context size only.

**Increasing the context size.** We show the effect of increasing the context size on localization and image generation performance. The image generation performance (measured with PSNR) and the localization accuracy (median euclidean distance between the predicted camera position and the ground truth) are shown in Fig. 6. The results were computed on all scenes from the test set.

We can see that the performance of both novel view synthesis and camera pose estimation increases with more context views. The change is most prominent in the first five views, but after that it keeps increasing as well.

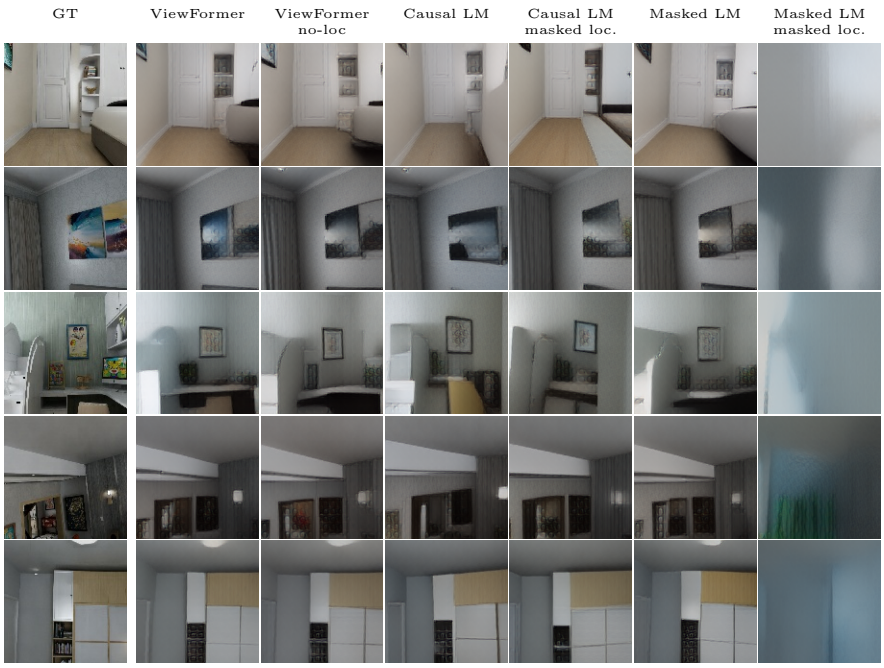


**Fig. 5.** Examples generated by alternative architectures described in Sec. D. The examples were generated on the test set of the InteriorNet dataset using context size 19.

**Table 2.** Ablation study evaluated on the InteriorNet dataset [16]. See Sec. D for a description of the compared variants. We show the PSNR, the pixel-wise MAE, and the LPIPS distance [36]. For localization, we show the median position error in meters and the median orientation error in degrees computed over all scenes.

| | **Image generation** | | | **Localization** |
|---|---|---|---|---|
| Method | PSNR↑ | MAE↓ | LPIPS↓ | Pos/Ori↓ |
| **ViewFormer** | 18.53 | 23.35 | 0.33 | **0.19**/**4.22** |
| ViewFormer no-loc | **19.10** | **21.56** | **0.32** | - |
| ViewFormer no-imagen | - | - | - | **0.19**/4.34 |
| Causal LM | 16.75 | 29.88 | 0.39 | - |
| Causal LM + masked loc. | 16.67 | 30.22 | 0.39 | 0.22/6.24 |
| Masked LM | 18.76 | 22.91 | **0.32** | - |
| Masked LM + masked loc. | 14.51 | 42.89 | 0.51 | 0.32/29.65 |

## E ShapeNet evaluation

In this section, we give more details on the ShapeNet results from the main paper (Fig. 7). We include quantitative and additional qualitative results. We trained our model on ShapeNet dataset rendered by SRN [29]. The context size used for training was three. We compare ViewFormer with SRN [29] and PixelNeRF [35]. We show the PSNR and SSIM [33] averaged across color channels for both car and chair categories with one or two context views. The results are presented in

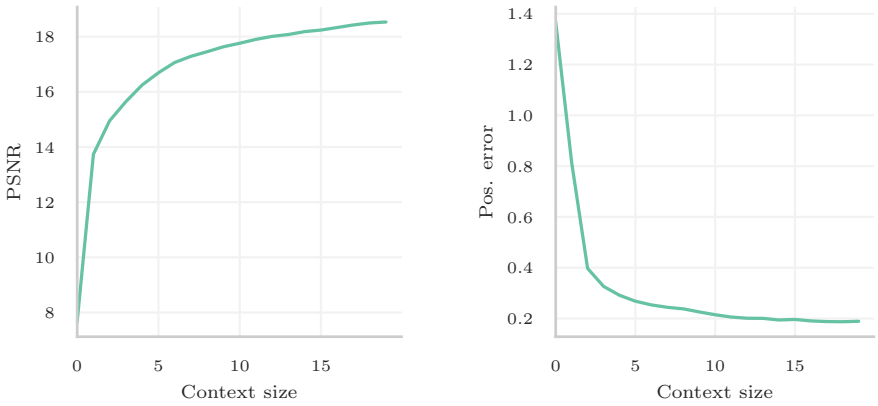


**Fig. 6.** This plot shows the effect of increasing the context size on the PSNR (**left**) and the position error (**right**) evaluated on the InteriorNet dataset [16]

**Table 3.** ShapeNet results comparing ViewFormer with SRN [29] and PixelNeRF [35]. We show the results for both car and chair category with one or two context views

| | | cars 1 view | | cars 2 views | | chairs 1 view | | chairs 2 views | |
|---|---|---|---|---|---|---|---|---|---|
| Method | 3D | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ |
| **ViewFormer** | ✗ | 19.03 | 0.83 | 20.09 | 0.85 | 14.74 | 0.79 | 17.20 | 0.84 |
| SRN [29] | ✓ | 22.25 | 0.89 | 24.84 | 0.92 | 22.89 | 0.89 | 24.48 | 0.92 |
| PixelNeRF [35] | ✓ | 23.72 | 0.91 | 26.20 | 0.94 | 23.17 | 0.90 | 25.66 | 0.94 |

Tab. 3. We also extend Fig. 7 from the paper with additional qualitative results on cars and chairs in Fig. 7 and 8.

From the results, we can see that our method performs worse than both SRN [29] and PixelNeRF [35] in terms of the quantitative results. This is expected because our method was designed for more views (more than 10) and was evaluated using one or two views. However, compared to PixelNeRF our method is able to recover more detail, whereas PixelNeRF produces blurry output, especially on the car category. Based on the qualitative results, we argue that although our approach has worse quantitative numbers, our results look more realistic. A possible cause for this observation could be that blurring the edges of an object can hide the unprecise geometry rendered by the model and increase PSNR. However, it loses fine detail in the images.

## F Shepard-Metzler-Parts-7 evaluation

We evaluated our model on the Shepard-Metzler-Parts-7 dataset [10,28] to compare our approach to other methods that only operate in 2D [6,10,30]. For the evaluation, we used the context size three. The additional qualitative results,


**Fig. 7.** Additional **ShapeNet** cars qualitative comparison with PixelNeRF [35] using two context views

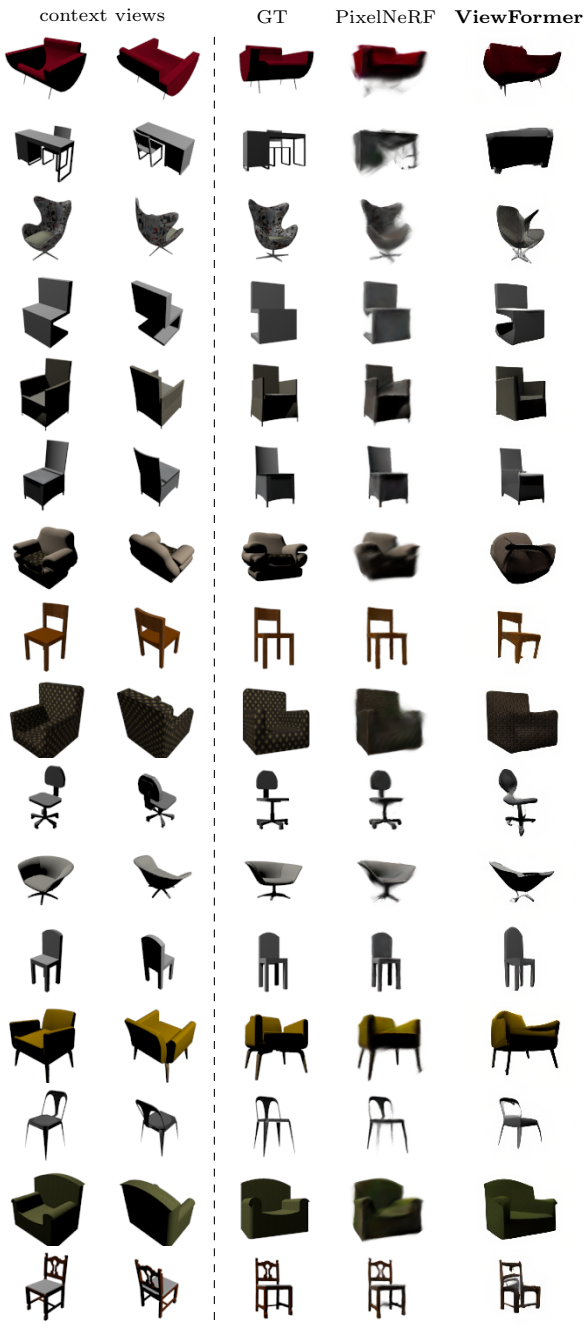


**Fig. 8.** Additional **ShapeNet** chairs qualitative comparison with PixelNeRF [35] using two context views

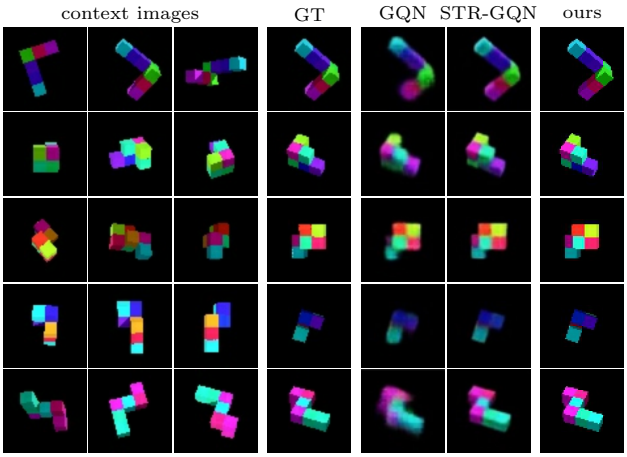


**Fig. 9.** Qualitative results on the SM7 dataset [10]. We compare against GQN [10] and STR-GQN [6]

**Table 4.** Comparison with GQN-based methods [5,10,30] on the SM7 dataset. We show the **MAE**, **RMSE**, and the position and orientation errors (**Pos**, **Ori**)

| | **Image generation** | | **Localization** |
|---|---|---|---|
| Method | MAE↓ | RMSE↓ | Pos/Ori↓ |
| **ViewFormer** | **1.61** | 7.02 | 0.21/3.48 |
| GQN [10] | 3.13 | 9.97 | - |
| E-GQN [30] | 2.14 | **5.63** | - |
| STR-GQN [5] | 3.11 | 10.56 | - |

presented in Fig. 9, extend Fig. 5 from the main paper. Unfortunately, in the qualitative analysis, we cannot compare with E-GQN [30] because the authors did not make the generated images or models public.

Tab. 4 presents quantitative results (averaged over 1000 scenes). As our method uses images of sizes $128 \times 128$ pixels, we rescaled the images before training the codebook. For evaluation, we used the original image size $64 \times 64$ pixels of the dataset. We report the pixel-wise mean absolute error (MAE) and root mean square error (RMSE). For reference, we also show the localization accuracy. The position error (Pos) is the median euclidean distance between the predicted positions and the ground-truth camera positions, and the orientation error (Ori) is the median of the angular distances in degrees.

As can be seen, our method clearly outperforms the baselines in terms of the MAE. E-GQN performs best in terms of the RMSE as it is trained to optimize this metric, whereas our method uses MAE and perceptual loss.

**Table 5.** Codebook evaluation on the SM7 [10,28], InteriorNet [16], CO3D [23], and 7-Scenes [12] datasets. We report the PSNR, MAE, and LPIPS metrics averaged over 1000 sampled images. The codebooks were evaluated with image size $128 \times 128$, except for 'CO3D@400', which was evaluated with image size $400 \times 400$ pixels

| dataset | PSNR↑ | MAE↓ | LPIPS↓ |
|---|---|---|---|
| SM7 | 36.96 | 1.06 | 0.0075 |
| InteriorNet | 24.86 | 11.01 | 0.1966 |
| CO3D | 25.14 | 5.70 | 0.0994 |
| CO3D@400 | 25.34 | 5.63 | 0.1670 |
| 7-Scenes (fine-tuned) | 19.29 | 17.51 | 0.2937 |
| 7-Scenes | 19.00 | 19.22 | 0.3621 |
| ShapeNet-cars | 23.50 | 5.46 | 0.0734 |
| ShapeNet-chairs | 27.43 | 2.75 | 0.0425 |

## G Codebook evaluation

In this section, we add more details on the codebook's representation capabilities (see Fig. 4 in the main paper) by showing quantitative results. We evaluated the codebook models on each dataset's test set. We report the peak signal-to-noise ratio (PSNR), mean absolute error computed in the RGB image space (MAE), and the LPIPS distance [36]. All codebooks were evaluated with image size $128 \times 128$ pixels except for 'CO3D@400', which was evaluated with image size $400 \times 400$ pixels to be comparable with [23]. The metrics are averaged over 1000 randomly sampled images. The results can be seen in Tab. 5.

Before training the final codebook, we experimented with different codebook models. We also trained the DALL·E codebook [22], which yielded slightly blurry images even when we used a codebook of size 8192 (normally, we use a codebook of size 1024). We observed a similar outcome with our codebook when we did not use the perceptual loss. We also tried to use a GAN loss for the codebook, as described in [11]. However, the generated images did not look geometrically consistent.

## H Training details

To allow our results to be reproduced, we give the details on the architecture of our method as well as the training hyperparameters.

All our **codebook models** were trained using the same set of hyperparameters.[5] We trained codebooks of size 1024. The architecture is very similar to [11] and is summarized in Sec. I. We used the Adam optimizer [15] with learning rate[6] $1.584 \times 10^{-3}$ to train for 200k steps (roughly 480 GPU-hours) with a batch size of 352. For the CO3D dataset, we trained on the same 10 object categories as

[5] Except for the SM7 dataset, where we only fine-tuned an existing model.
[6] The learning rate was rescaled from prior experiments; $1.6 \times 10^{-3}$ would work too.

in [23] as well as on the full dataset. For the 7-Scenes dataset, due to not having enough images to train from scratch, we finetuned an InteriorNet pre-trained model. Therefore, we used only 20k batch updates with the same hyperparameters.

The architecture of our **transformer model** is based on GPT2-base [20], and has 12 transformer blocks, 12 attention heads, and the hidden size is 768. The model design was chosen based on its successes in other domains and because its size fits well on our hardware. We trained our transformer models using the AdamW optimizer [17]; we used the cosine schedule for the learning rate with a 2k step linear warmup.

For the **InteriorNet dataset**, we used the mixed-precision training with learning rate $8 \times 10^{-5}$, batch size 40, and learning rate decay 0.01. The context size was 19, but we did not optimize the first four views. The weight of the localization loss term was 5. In all other experiments, the localization loss weight was 1 unless stated otherwise.

For the **Shepard-Metzler-7-Parts (SM7)** [10,28] dataset, we trained the transformer for 120k steps with the context size 5, batch size 128, and the learning rate $10^{-4}$ (cosine decay, warmup). Before passing camera poses into the transformer, we normalized the positions by multiplying them by 0.2. We also gradually increased the weight of the localization term from 0 to 1 using the cosine schedule.[7]

For the **CO3D dataset**, we fine-tuned the model trained on the InteriorNet dataset. For the 10 categories, we optimized the model for 40k gradient steps with learning rate $10^{-4}$ (cosine decayed with a 2,000 step warmup), weight decay 0.05, and batch size 80, employing mixed-precision training. The context size was 9, and the batch size was 80. We scaled the camera positions by 0.05 in order for the positions to have a similar range as the pre-trained model. We also trained a model on all dataset categories using 100k gradient steps with the batch size 40, without using mixed-precision training, and when using the localization, we further used gradient clipping with the norm 1 to improve stability.

For the **7-Scenes dataset**, we used a single InteriorNet pre-trained model which we fine-tuned on all 7-Scenes scenes. Same as in the original model, the context size was 19, but we did not optimize the first four views. The transformer was fine-tuned for 10k gradient steps with learning rate $10^{-5}$ (cosine schedule, warmup). We rescaled the positions by multiplying them by 5 to be in the same range as InteriorNet.

Finally, for the **ShapeNet dataset**, we fine-tuned InteriorNet pre-trained model as well. We trained a single model for both categories: cars and chairs with the context size 3. We did not use mixed-precision training and the batch size was 64. The transformer was fine-tuned for 100k gradient steps with learning rate $10^{-4}$ (cosine schedule, warmup), weight decay was 0.05, and we used gradient clipping with the norm 1.

[7] The schedule is not needed for the training to work and in newer experiments, we use a constant instead.

**Table 6.** Codebook architecture details: the encoder (**top left**), the decoder (**right**), and the residual block (**bottom left**). For each layer, we list the number of output features (Num. features) and their sizes (Out. size). We denote kernel size as 'ks', stride as 's', and the number of groups as 'g'. We use nearest neighbor for the Upsample 2D layer. Note that the output of the residual block is added to its input as in ResNets [13]. If the number of input channels is not equal to the number of output channels, the residual connection is implemented by applying an affine transformation to the input features position-wise before summing them with the output of this block

| Layer type | Num. features | Out. size |
|---|---|---|
| Conv 2D (ks: 3) | 128 | 128 |
| ResBlock | 128 | 128 |
| ResBlock | 128 | 128 |
| Conv 2D (ks: 3, s: 2) | 128 | 64 |
| ResBlock | 128 | 64 |
| ResBlock | 128 | 64 |
| Conv 2D (ks: 3, s: 2) | 128 | 32 |
| ResBlock | 256 | 32 |
| ResBlock | 256 | 32 |
| Conv 2D (ks: 3, s: 2) | 256 | 16 |
| ResBlock | 256 | 16 |
| Attention 2D | 256 | 16 |
| ResBlock | 256 | 16 |
| Attention 2D | 256 | 16 |
| Conv 2D (ks: 3, s: 2) | 256 | 8 |
| ResBlock | 512 | 8 |
| ResBlock | 512 | 8 |
| ResBlock | 512 | 8 |
| Attention 2D | 512 | 8 |
| ResBlock | 512 | 8 |
| GroupNorm 2D [34] (g: 32) | 512 | 8 |
| Swish [21] | 512 | 8 |
| Conv 2D (ks: 3) | 256 | 8 |
| Conv 2D (ks: 1) | 256 | 8 |

(a) Encoder

| Layer | Num. features |
|---|---|
| GroupNorm [34] (g: 32) | in |
| Swish [21] | in |
| Conv 2D (ks: 3) | out |
| GroupNorm [34] (g: 32) | out |
| Swish [21] | out |
| Conv 2D (ks: 3) | out |

(b) ResBlock

| Layer type | Num. features | Out. size |
|---|---|---|
| Conv 2D (ks: 1) | 256 | 8 |
| Conv 2D (ks: 3) | 512 | 8 |
| ResBlock | 512 | 8 |
| Attention 2D | 512 | 8 |
| ResBlock | 512 | 8 |
| ResBlock | 512 | 8 |
| ResBlock | 512 | 8 |
| ResBlock | 512 | 8 |
| Upsample 2D | 512 | 16 |
| Conv 2D (ks: 3) | 512 | 16 |
| ResBlock | 256 | 16 |
| Attention 2D | 256 | 16 |
| ResBlock | 256 | 16 |
| Attention 2D | 256 | 16 |
| ResBlock | 256 | 16 |
| Attention 2D | 256 | 16 |
| Upsample 2D | 256 | 32 |
| Conv 2D (ks: 3) | 256 | 32 |
| ResBlock | 256 | 32 |
| ResBlock | 256 | 32 |
| ResBlock | 256 | 32 |
| Upsample 2D | 256 | 64 |
| Conv 2D (ks: 3) | 256 | 64 |
| ResBlock | 128 | 64 |
| ResBlock | 128 | 64 |
| ResBlock | 128 | 64 |
| Upsample 2D | 128 | 128 |
| Conv 2D (ks: 3) | 128 | 128 |
| ResBlock | 128 | 128 |
| ResBlock | 128 | 128 |
| ResBlock | 128 | 128 |
| GroupNorm 2D [34] (g: 32) | 128 | 128 |
| Swish [21] | 128 | 128 |
| Conv 2D (ks: 3) | 128 | 3 |

(c) Decoder

# I Codebook architecture

In Tab. 6 we give the details on the codebook architecture (*cf.* Sec. 3 in the main paper). The codebook model architecture was taken from [11] and modified slightly to downscale the images into two times smaller latent space. We have chosen this architecture because it had shown promising results for image generation in combination with transformers [11]. The other architecture we had considered was DALL·E [22], but from our experiments, it performed worse.

## References


1. Balntas, V., Li, S., Prisacariu, V.: RelocNet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 751–767 (2018) 5, 7
2. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 5
3. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2021) 7
4. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2616–2625 (2018) 5, 7
5. Chen, S., Wang, Z., Prisacariu, V.: Direct-posenet: Absolute pose regression with photometric consistency. arXiv preprint arXiv:2104.04073 (2021) 13
6. Chen, W.C., Hu, M.C., Chen, C.S.: STR-GQN: Scene representation and rendering for unknown cameras based on spatial transformation routing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5966–5975 (2021) 10, 13
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019) 8
8. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: CamNet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2871–2880 (2019) 5, 7
9. Engilberge, M., Collins, E., Susstrunk, S.: Color representation in deep neural networks. In: Proceedings of the IEEE International Conference on Image Processing. pp. 2786–2790 (2017) 5
10. Eslami, S.A., Rezende, D.J., Besse, F., Viola, F., Morcos, A.S., Garnelo, M., Ruderman, A., Rusu, A.A., Danihelka, I., Gregor, K., et al.: Neural scene representation and rendering. Science **360**(6394), 1204–1210 (2018) 10, 13, 14, 15
11. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12873–12883 (2021) 8, 14, 16
12. Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time RGB-D camera relocalization. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 173–179. IEEE (2013) 1, 5, 6, 14
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 16
14. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2938–2946 (2015) 5, 7
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (Poster) (2015) 14

16. Li, W., Saeedi, S., McCormac, J., Clark, R., Tzoumanikas, D., Ye, Q., Huang, Y., Tang, R., Leutenegger, S.: InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In: British Machine Vision Conference (BMVC) (2018) 1, 2, 7, 8, 9, 10, 14
17. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018) 15
18. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: LENS: Localization enhanced by NeRF synthesis. In: 5th Annual Conference on Robot Learning (2021) 5, 7
19. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: International Conference on Machine Learning. pp. 4055–4064. PMLR (2018) 8
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019) 8, 15
21. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017) 16
22. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. arXiv preprint arXiv:2102.12092 (2021) 8, 14, 16
23. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10901–10911 (2021) 1, 2, 3, 14, 15
24. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: CVPR (2019) 5, 7
25. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(9), 1744–1756 (2016) 5, 7
26. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: Proceedings of the IEEE/CVF Conference On computer Vision and Pattern Recognition. pp. 3302–3312 (2019) 5, 7
27. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. arXiv preprint arXiv:2103.11468 (2021) 5, 7
28. Shepard, R.N., Metzler, J.: Mental rotation of three-dimensional objects. Science **171**(3972), 701–703 (1971) 10, 14, 15
29. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3D-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019) 9, 10
30. Tobin, J., Zaremba, W., Abbeel, P.: Geometry-aware neural rendering. Advances in Neural Information Processing Systems **32**, 11559–11569 (2019) 10, 13
31. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1808–1817 (2015) 5
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 8
33. Wang, Z., Bovik, A.C.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. IEEE signal processing magazine **26**(1), 98–117 (2009) 9
34. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) 16

35. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021) 2, 9, 10, 11, 12
36. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–595 (2018) 9, 14