ViewNeRF: Unsupervised Viewpoint Estimation Using Category-Level Neural Radiance Fields - Supplementary Material

Octave Mariotti
omariott@ed.ac.uk
Oisin Mac Aodha
oisin.macaodha@ed.ac.uk
Hakan Bilen
hbilen@ed.ac.uk

School of informatics
University of Edinburgh
Edinburgh, UK

A Pose regularization

We provide pseudo-code for our pose regularization method. Note that K might not be equal to B. In practice, instead of using the minimal distance, we use a soft minimum to decrease noise. The pose prior approximately follows the training data distribution, i.e. top half of the sphere on NeRF scenes, uniform on ShapeNet, ground level for Freiburg cars. Regularization strength λ starts at 1 and undergoes exponential scheduling, being multiplied by 0.1 every 10 epochs before being turned off at epoch 30.

Algorithm 1: Pose regularization

```
Input: Minibatch of predicted poses $p^*_{1,\dots,B}$, Prior distribution $\mathcal{P}$, number of
       samples $K$
Output: Regularization loss $\mathcal{L}_{reg}$
$\mathcal{L}_{reg} = 0$
for $i \in 1 \dots K$ do
    $p' \sim \mathcal{P}$ ;                    // draw a pseudo-target from $\mathcal{P}$
    dists $= ||p^* - p'||$ ;   // distance between each predicted pose
     and p', size $B$
    weights $= SoftMax(-$dists$)$ ;                  // Batch-wise SoftMax
    weighted_dists $=$ weights $*$ dists
    $\mathcal{L}_{reg} += \frac{1}{K} * Avg($weighted_dists$)$ ;       // Batch-wise Average
```

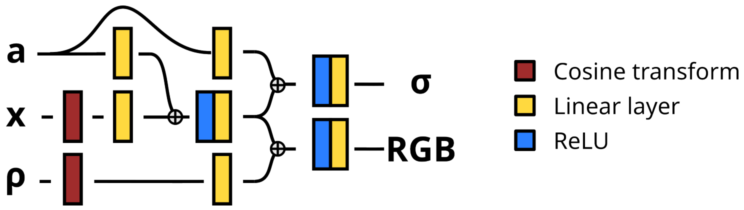

Figure 1: NeRF archtitecture used in ViewNeRF. **a** depicts the appearance embedding, while **x** and **ρ** are the spatial coordinate and viewing direction.

B Implementation details

Perceptual loss. We implement perceptual loss using a pretrained VGG16 model. The total loss consists of standard MSE in pixel space, plus MSE between features extracted before the first, second and third max polling layers of the model, with weights 10, 1, 1, and 1 respectively. Because of the cost of producing full images, the output reconstruction uses a 64x64 resolution.

NeRF architecture. The architecture of our NeRF decoder is depicted in Fig. 1. To encourage 3D consistency, we use cosine embeddings of size 8 and 1 for **x** and **ρ** respectively. They are then mapped with linear layers to the inner dimension of the model which is 128.

C Supplemental results

C.1 Reconstruction metrics

While ViewNeRF is not designed for accurate reconstructions, quantitative values could be useful for future references. SSIM and PSNR for ShapeNet are shown in Table 1a.

C.2 Extra visualizations

In Fig. 2, we provide extra comparison between our method and ViewNet on Freiburg cars, by sampling views at a 45° interval around reconstructed test instances. It is apparent that ViewNet does not manage to reconstruct the back of the car correctly.

D Speed analysis

Table 1b shows the time taken to process 1 64x64 image on a Tesla V100. The inference time of CodeNeRF, requiring 300 gradient descent steps, is 5 orders of magnitude higher than ours, while its larger NeRF backbone also makes it more expensive during training.

	ShapeNet cars	ShapeNet chairs	Freiburg cars
PSNR (↑)	15.6	17.5	19.1
SSIM (↑)	0.70	0.77	0.84

(a) ViewNeRF reconstruction metrics on evaluation instances.

	CodeNeRF	ViewNet	ViewNeRF
Train	139 ± 9.6	9.1 ± 3.7	14.7 ± 1.7
Inference	$38.4 \pm 0.6 \times 10^3$	0.1 ± 0.3	0.4 ± 1.0

(b) Processing time per sample in milliseconds (ms), reported as mean and standard deviation averaged over 100 batches.

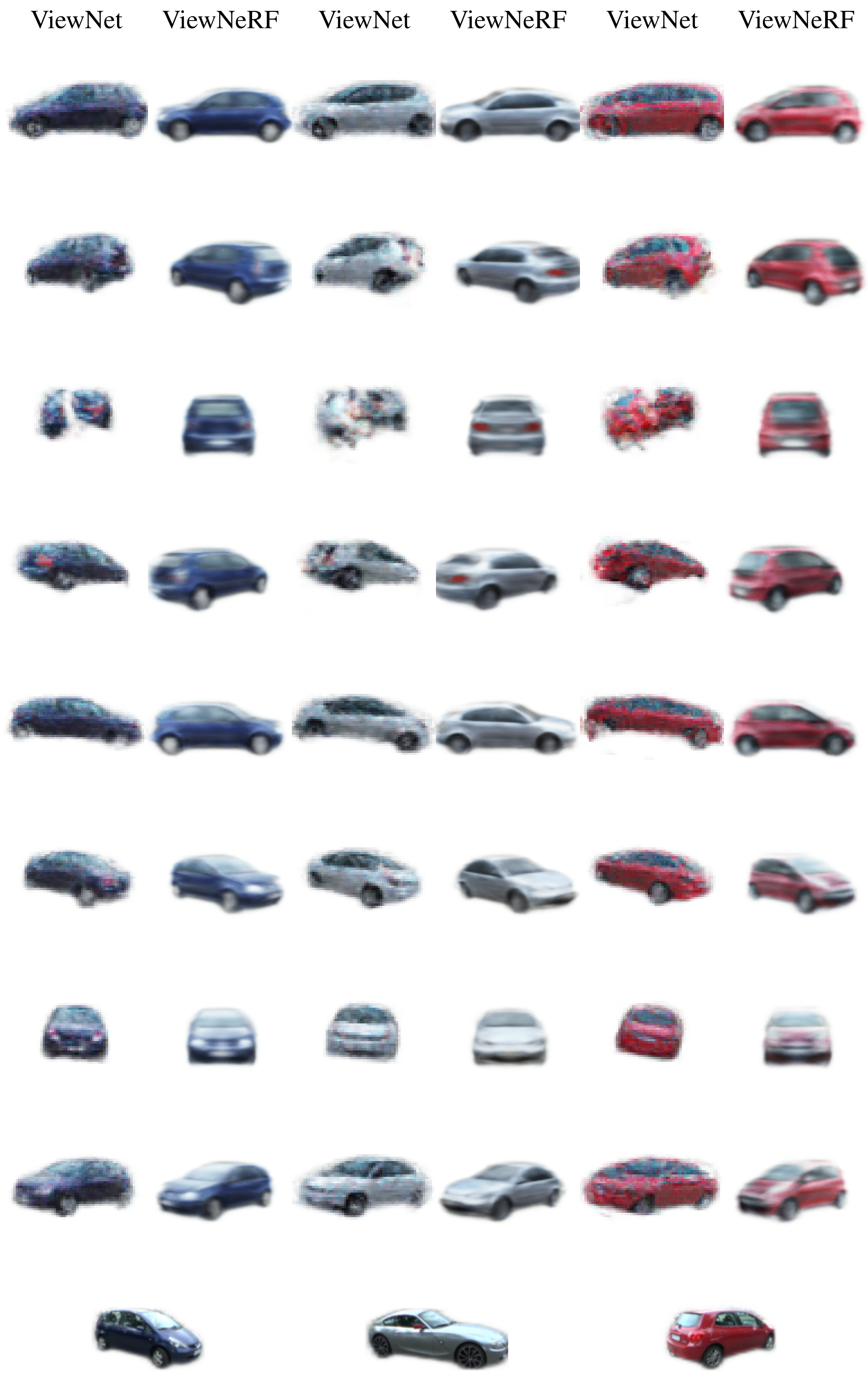

Figure 2: Reconstructions of Freiburg car test instances for ViewNet and ViewNeRF. Bottom row is the frame used for providing appearance embedding.