Video Autoencoder: self-supervised disentanglement of static 3D structure and motion

Zihang Lai
Carnegie Mellon University

Sifei Liu
Nvidia

Alexei A. Efros
UC Berkeley

Xiaolong Wang
UC San Diego

1. Additional View Synthesis Results

1.1. Our Qualitative Results

Figure 1, 2 and 3 provide additional novel view synthesis results. Our method is able to synthesize high quality view with correct camera transformation.

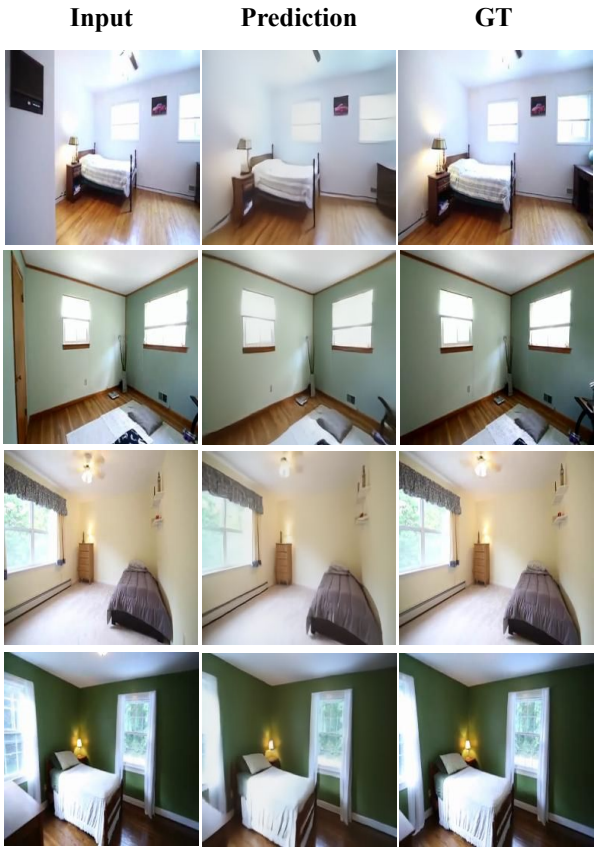

Figure 1: **Novel View Synthesis:** Our method is able to generate accurate results that correspond to the ground truths. Our method works for both camera translation (row 1-2) and rotation (row 3-4).

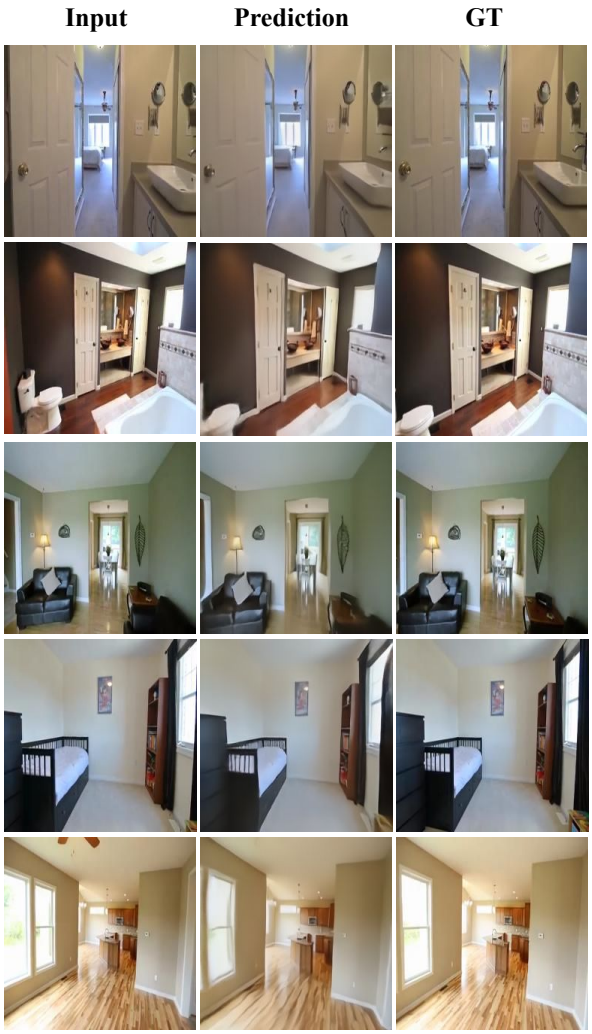

Figure 2: **Novel View Synthesis (continued)**

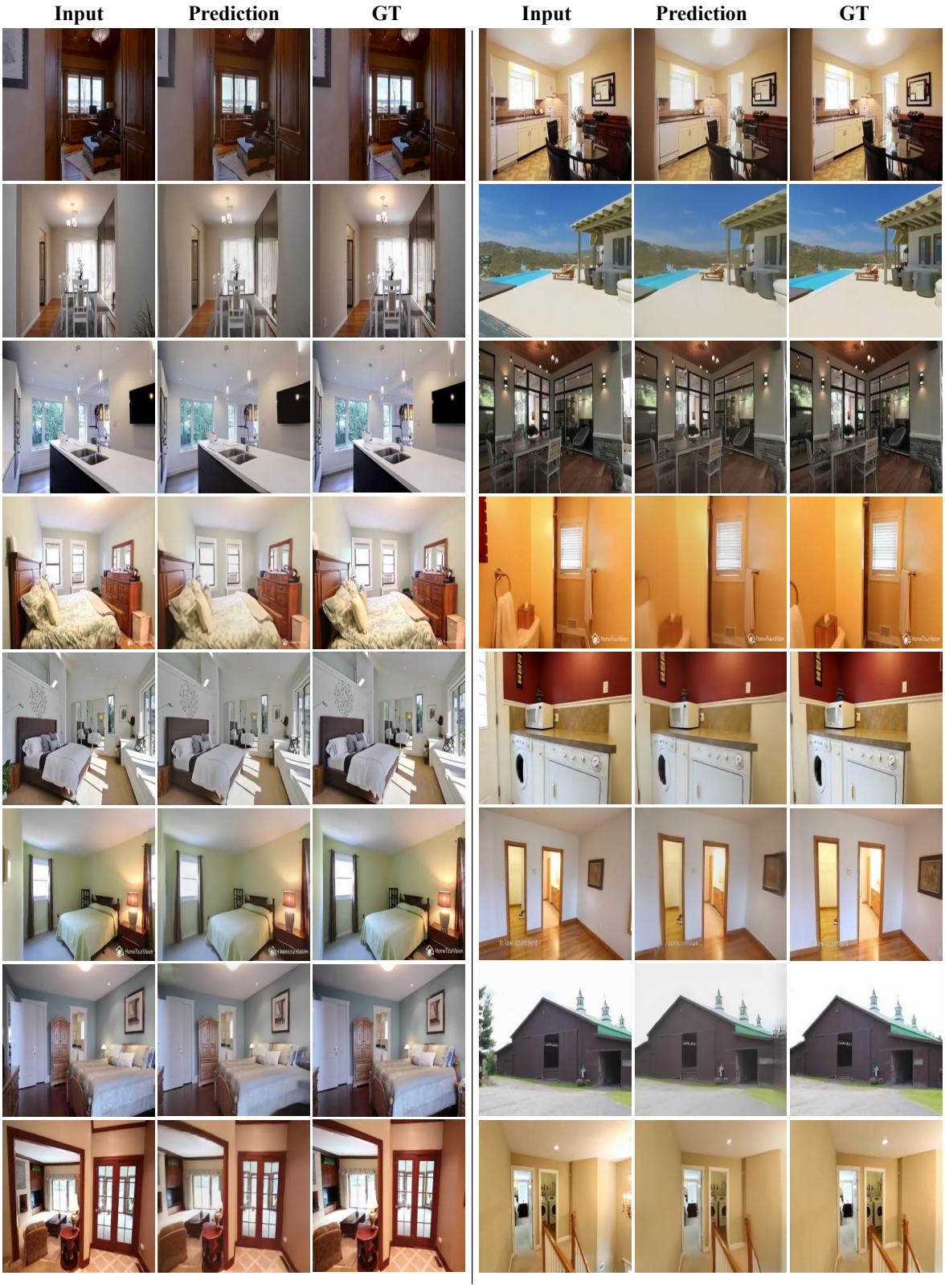

Figure 3: **Novel View Synthesis (continued)**

1.2. Qualitative Comparison with Previous Methods

Figure 4 and 5 provide additional comparison with previous methods. In Fig. 6, we show additional detailed comparison with the strong competitor, Synsin [1]. Our method is able to generate more clear and accurate results, even though our method is trained without any camera supervision.

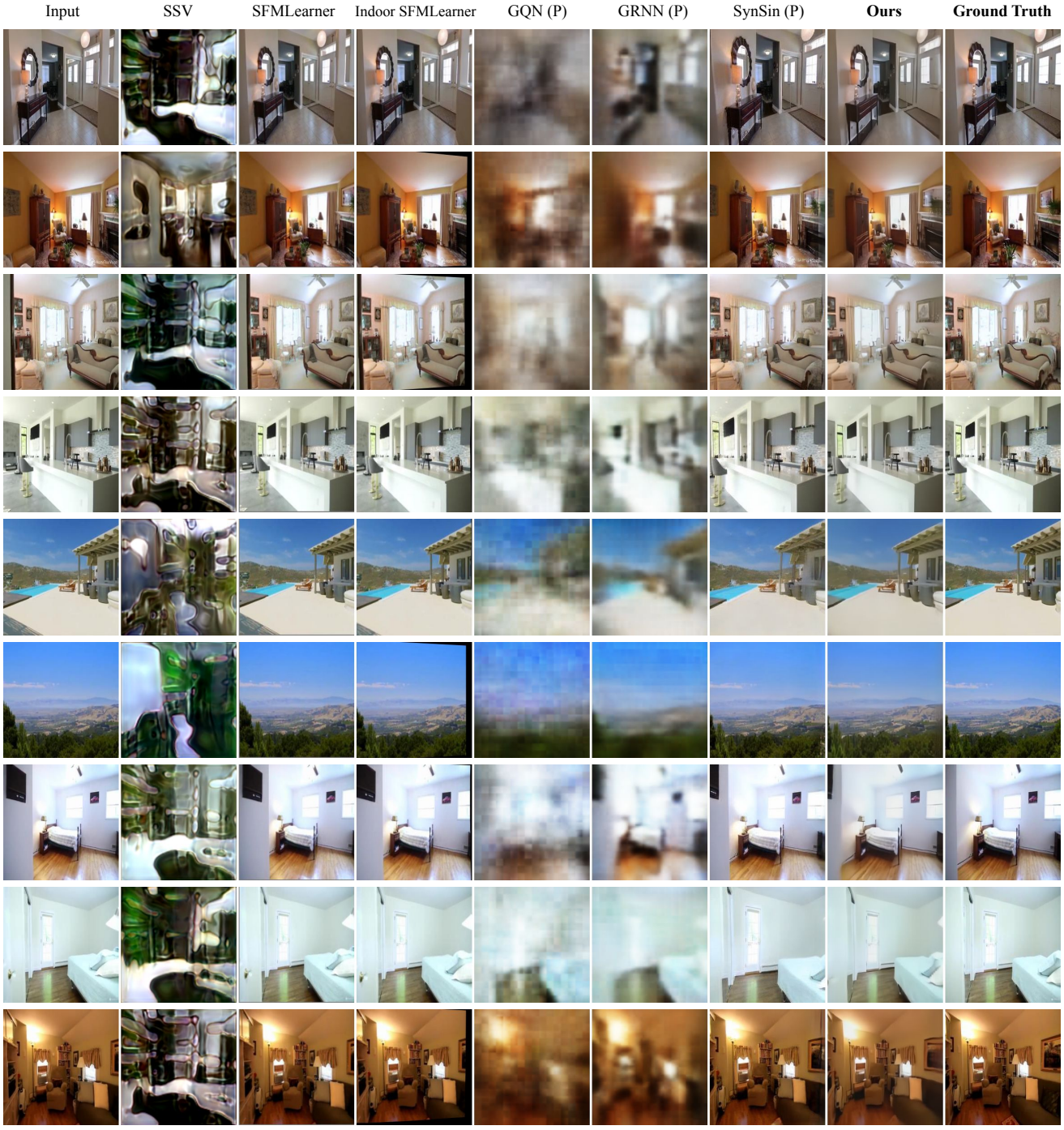

Figure 4: **Novel view synthesis results compared with previous methods:** Other methods show systematic errors either in rendering or pose estimation. Our method is able to outperform other unsupervised methods visually by large margin. (P) indicates methods supervised by true camera transformations.

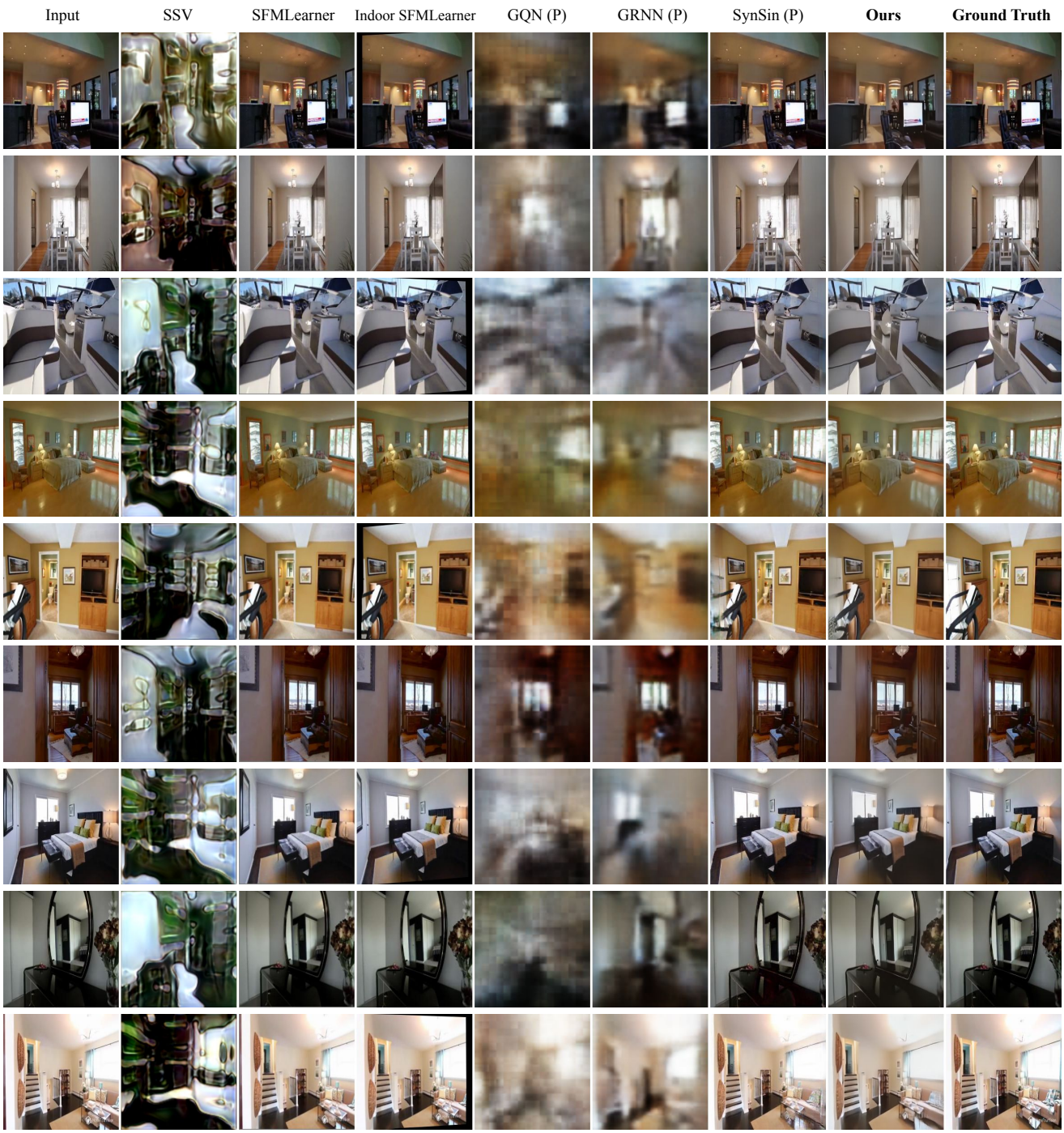

Figure 5: **Novel view synthesis results compared with previous methods (continued)**

2. Network structures

In Table 1,2,3, we give additional information about the network strucutres of the subcomponents of our model. There is a total of 29M trainable parameters, mainly in the Resnet-50 feature extractor.

3. Details of baselines

In this section, we describe the baselines we used to compare our method with. Specifically, we compared with Dosovitsky et al. [2], Appearance Flow [3], StereoMag [4], Synsin [1], SFMLearner [5], Indoor SFM-

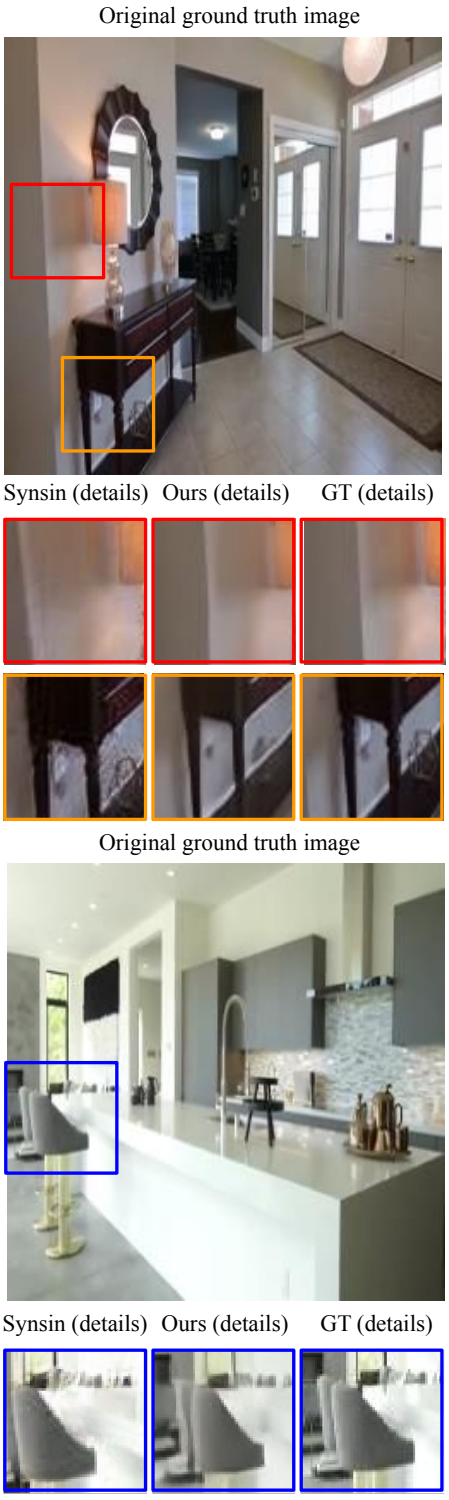

Figure 6: **Novel view synthesis results compared with [1] (details)** Our method synthesizes novel views with less artifacts and noises. See Fig. 4 for uncropped full results of our model and [1].

Learner [6], GQN [7], GRNN [8] and SSV [9].

Dosovitsky et al. This method infers a novel view of a given input with a neural network. The model feeds an image and the desired viewpoint into the network, which directly estimates an image corresponding to the given view. This method was only tested on images of a single object, which are considerably easier than the real scene data we used.

Appearance Flow. Appearance flow predicts a flow field that warps the original image into a novel view. This flow field is computed from a ConvNet which takes the original image and the viewpoint transformation as input. This method is shown to work very well on synthetic object datasets (*e.g. ShapeNet [10]*). It has also shown that the performance on real-world scenes surpasses direct pixel prediction [11].

SynSin. Synsin is a novel view synthesis algorithm that aims to synthesize new viewpoints of a given single image. The algorithm makes use of an intermediate representation: a point cloud where each point is a feature vector. Taking a single image as input, Synsin first computes a depth map and a set of 2D feature maps. The 2D feature maps are projected back into 3D points with the depth map. Next, a camera transformation is given and the point cloud is rendered with the new camera pose. There are two major differences in the high-level philosophy between our model and Synsin. First, we do not require true camera transformation during training time. This means that Video Autoencoder works on raw videos and Synsin can only be trained on datasets with camera pose obtained from sensors or precomputed from SfM. Second, Synsin leverages explicit 3D representation (*i.e.* the depth map) whereas the Video Autoencoder does not. As shown in the paper, explicit 3D representation could fail in out-of-domain data whereas our method generalizes better into unseen images.

StereoMag. StereoMag constructs multiplane images (MPIs) from two input views. The multiplane images represent a 3D structure as a set of fronto-parallel planes at fixed depths. It can be rendered at a given viewpoint by *blending* planes with a set of blending weights. These blending weights are predicted from one image, and the other image is used as the plane sweep volume (*i.e.* the value of the planes). Apart from the representation, one major difference between our method and StereoMag is that StereoMag makes use of two images as input, which greatly simplifies the problem.

SFMLearner. SFMLearner disentangle a short video clip into depth maps and camera trajectory. The method is structurally similar to our method. However, there are several major differences. (i) SFMLearner (and similarly, GeoNet, etc.) results were shown on the relatively simple KITTI dataset, but work poorly on more complex data [12]. (ii) SFMLearner requires ground truth camera intrinsics, which makes it more difficult to train on raw videos. (iii) SFM-

Stage	Configuration	Output
0	Input image	$H \times W \times 3$
	2D feature extraction	
1	Extract feature with Resnet-50	$\frac{H}{16} \times \frac{W}{16} \times 2048$
	Reshaping 2D to 3D	
2	Reshape feature dimension to 256	$\frac{H}{16} \times \frac{W}{16} \times 8 \times 256$
	3D Convolutions	
3	3D-Deconvolution (4^3 kernel, 128 filters, stride 2)	$\frac{H}{8} \times \frac{W}{8} \times 16 \times 128$
4	3D-Deconvolution (4^3 kernel, 32 filters, stride 2)	$\frac{H}{4} \times \frac{W}{4} \times 32 \times 32$

Table 1: 3D Encoder ($\mathcal{F}_{3D}$) architecture.

Stage	Configuration	Output
0	Two concatenated input image	$H \times W \times 6$
	2D feature extraction	
1	2D-convolution (3^2 kernel, 16 filters, stride 2)	$\frac{H}{2} \times \frac{W}{2} \times 16$
2	2D-convolution (3^2 kernel, 32 filters, stride 2)	$\frac{H}{4} \times \frac{W}{4} \times 32$
3	2D-convolution (3^2 kernel, 64 filters, stride 2)	$\frac{H}{8} \times \frac{W}{8} \times 64$
4	2D-convolution (3^2 kernel, 128 filters, stride 2)	$\frac{H}{16} \times \frac{W}{16} \times 128$
5	2D-convolution (3^2 kernel, 256 filters, stride 2)	$\frac{H}{32} \times \frac{W}{32} \times 256$
6	2D-convolution (3^2 kernel, 256 filters, stride 2)	$\frac{H}{64} \times \frac{W}{64} \times 256$
7	2D-convolution (3^2 kernel, 256 filters, stride 2)	$\frac{H}{128} \times \frac{W}{128} \times 256$
8	2D-convolution (1^2 kernel, 6 filters, stride 1)	$\frac{H}{128} \times \frac{W}{128} \times 6$
9	Mean pooling	6
10	Multiply by 0.01	6

Table 2: Architecture of ConvNet ($\mathcal{H}$) for Trajectory Encoder ($\mathcal{F}_{Traj}$). The last step could stabilize the training process.

Learner cannot produce satisfactory view synthesis results because its spatial representation is a 2.5D depth map. We use the predicted depth and camera transformation to warp the first frame into the target frame.

Indoor SFMLearner. P^2-Net, or the Indoor SFM-Learner, proposes to use a patch-based loss to overcome the optimization problem of SFMLearner, and shows better results in indoor scenes. Similar to SFMLearner, we use predicted depth and camera to warp the first frame into the target frame. Because not all pixels in the target frame have a corresponding pixel in the first frame, this method could produce large blank areas (as seen in Fig. 4 and 5).

GQN Generative Query Network combines 2D features of multiple input images into a 2D *Neural Scene Representation*. This 2D representation is then decoded with an LSTM network conditioned on a query viewpoint. Although GQN is shown to yield good results on toy datasets, we find it

Stage	Configuration	Output
0	Input 3D deep voxels	$\frac{H}{4} \times \frac{W}{4} \times 32 \times 32$
1	Input 3D transformation	6
	Rotating 3D deep voxels	
2	Rotate deep voxels with input transf.	$\frac{H}{4} \times \frac{W}{4} \times 32 \times 32$
3	3D-Convolution (3^3 kernel, 64 filters, stride 1)	$\frac{H}{4} \times \frac{W}{4} \times 32 \times 64$
4	3D-Convolution (3^3 kernel, 64 filters, stride 1)	$\frac{H}{4} \times \frac{W}{4} \times 32 \times 64$
	Reshape 3D to 2D	
5	Concatenate feature and depth dim.	$\frac{H}{4} \times \frac{W}{4} \times 2048$
	2D Convolutions (neural network renderer)	
6	2D-convolution (1^2 kernel, 512 filters, stride 1)	$\frac{H}{4} \times \frac{W}{4} \times 512$
7	2D-Deconvolution (4^2 kernel, 64 filters, stride 2)	$\frac{H}{2} \times \frac{W}{2} \times 64$
8	2D-Deconvolution (4^2 kernel, 32 filters, stride 2)	$H \times W \times 32$
9	2D-Deconvolution (3^2 kernel, 3 filters, stride 1)	$H \times W \times 3$

Table 3: Decoder ($\mathcal{G}$) architecture.

struggles to generate clear results for real-world scenes.

GRNN GRNN constructs an RNN-aggregated 3D voxel from input images as scene representation. This 3D voxel could be projected back into a set of 2D feature maps with a specific query viewpoint and decoded into a query view. While GRNN also makes use of a 3D deep voxel as representation, several significant differences between our model and GRNN include: (i) GRNN only support 2 degrees of freedom for camera transformations (yaw and pitch) whereas we support a full 6 DOF camera transformation (ii) GRNN is trained with ground truth cameras whereas we do not make use camera supervision. (iii) GRNN requires a complicated matching process during testing. As shown in the main text and Fig. 4 and 5, GRNN fails to produce satisfactory results on RealEstate10K.

SSV. SSV is the state-of-the-art self-supervised viewpoint estimation algorithm. Leveraging multiple constraints such as cycle consistencies, SSV learns from image collections to predict camera poses of these images. We retrained SSV on the same training dataset, treating video frames as individual images. The output of SSV is a rotation angle of a single image. To obtain the relative pose, we concatenate pose predictions of the reference frame and current frame. We then fit a linear regression model to predict the true camera transformation using about 600 true poses, similar to SSV's original testing procedure. For fair comparisons, we also applied the Umeyama alignment [13].

4. Supplementary video

To better visualize the animated results of our model, we also provide a supplementary video `01710_supp.mp4`. In

the video, we show the results in videos rather than frames as in the paper. We also provide motivation and descriptions of our method.

References

[1] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "Synsin: End-to-end view synthesis from a single image," in *Proc. CVPR*, 2020. 3, 4, 5

[2] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *Proc. CVPR*, 2015. 4

[3] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proc. ECCV*, 2016. 4

[4] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *Proc. ACM SIGGRAPH*, 2018. 4

[5] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. CVPR*, 2017. 4

[6] Z. Yu, L. Jin, and S. Gao, "P^{2}net: Patch-match and plane-regularization for unsupervised indoor depth estimation," in *ECCV*, 2020. 5

[7] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, *et al.*, "Neural scene representation and rendering," *Science*, vol. 360, no. 6394, pp. 1204–1210, 2018. 5

[8] H.-Y. Fish Tung, R. Cheng, and K. Fragkiadaki, "Learning spatial common sense with geometry-aware recurrent networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5

[9] S. K. Mustikovela, V. Jampani, S. D. Mello, S. Liu, U. Iqbal, C. Rother, and J. Kautz, "Self-supervised viewpoint learning from image collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3971–3981, 2020. 5

[10] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 5

[11] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *European Conference on Computer Vision*, pp. 322–337, Springer, 2016. 5

[12] J. Zhou, Y. Wang, K. Qin, and W. Zeng, "Moving indoor: Unsupervised video depth learning in challenging environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8618–8627, 2019. 5

[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE PAMI*. 6