Appendix of Multi-view to Novel View: Synthesizing Novel Views with Self-Learned Confidence

The organization of the appendix is as follows. We present network architecture details as well as the implementation and training details in Section A and Section B. In Section C, we present additional results and intermediate predictions, along with qualitative results for the ablation study. In Section D, we study the effectiveness of confidence maps produced by our model. In Section E, we investigate how the source image ordering affects the synthesized results. In Section F, we include details on how pose information is fed in to the modules.

A Detailed Network Architectures

In Fig. 1, we show a diagram of our flow predictor, and in Fig. 2, we show a diagram of our recurrent pixel generator. For the deconvolutional layer, we upsample the input features using nearest neighbor interpolation and apply a convolution with stride 1. The pixel generator uses Convolutional LSTMs [1]. The equations for the convolutional LSTMs(convLSTM) are:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} \circ \mathcal{X}_t + W_{ho} \circ \mathcal{H}_{t-1} + W_{co} \circ c_t + b_o) \\
\mathcal{H}_t &= o_t \circ tanh(\mathcal{C}_t).
\end{aligned}
$$

For some arbitrary time step t, $\mathcal{X}_t$ denotes the feature maps encoded by the encoder, $\mathcal{C}_t$ denotes the cell outputs, and $\mathcal{H}_t$ denotes the hidden state. $\circ$ denotes Hadamard product, $*$ denotes convolution operation, and i_t, f_t, o_t denote gates. We used ConvLSTM inside our residual blocks; therefore, the the output of the ConvLSTM is $\mathcal{C}_t + \mathcal{X}_t$.

Discriminator architecture. Let $\mathcal{C}_{s,k,c}$ denote a convolutional layer with a stride s, kernel size k, and an output channel c. Then, the discriminator architecture can be expressed as $\mathcal{C}_{2,4,32} \rightarrow \mathcal{C}_{2,4,64} \rightarrow \mathcal{C}_{2,4,128} \rightarrow \mathcal{C}_{2,4,256} \rightarrow \mathcal{C}_{1,1,1}$. Note that we use a local discriminator similar to that of [2]. We use a Leaky ReLU activation function with slope of 0.2 on every layer, except for the last layer. Normalization layer is not applied. This architecture is shared across all experiments.

B Implementation and Training Details

We implemented our model on TensorFlow [3]. Our model is trained end-to-end using using ADAM optimization [4] with hyperparameters $\beta_1 = 0.9$ and

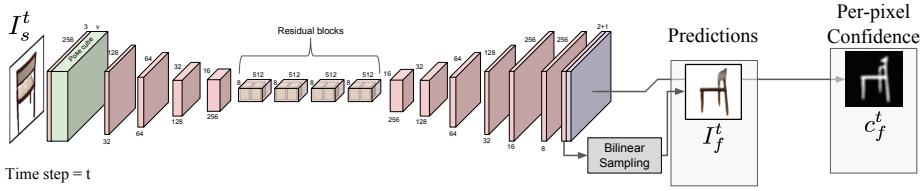

Fig. 1. The detailed architecture of the flow predictor.

$\beta_2 = 0.999$. We used a batch size of 8 for ShapeNet objects and 16 for KITTI and Synthia. The flow predictor is trained using a learning rate of 5×10^{-5} and the recurrent pixel generator is trained using a learning rate 10^{-4}.

C Additional Results

We provide additional results on ShapeNet car and chairs, KITTI, and Synthia. The ShapeNet results are generated using 4 source images (shown in Figure 3), while KITTI and Synthia results are generated using 2 source images (shown in Figure 4). In these figures, we also included the intermediate predictions synthesized by the flow predictor and the recurrent pixel generator.

D A Study of Predicted Confidence Maps

Confidence maps for a single example. We visualize the predicted confidence maps to understand how the confidence changes with respect to the target pose, (shown in Figure 5 and Figure 6). The confidence maps are generated from models trained on cars and chairs respectively. The column on the left is the target images and their poses, and the row on the top is the given source images and their poses. We observed that the model can reliably predict which regions would be visible from the source image; the smaller the disparity between the source and the target pose is, the more confident the model is.

Confidence matrices. To investigate how the predicted confidence values change with respect to the source and target pose, we collect all confidence maps predicted during the evaluation (20k testing tuples for each category). For each pair of source and target pose, we sum up the predicted confidence values across spatial dimensions (H and W) and average them across different objects. From Figure 7 (a) and Figure 7 (b), we can observe that the confidence maps have higher values when the target pose is close to the source pose, resulting in a brighter diagonal line. We also observed that the model learns to leverage the symmetry of the object.

Averaged confidence maps. We can also average the confidence maps across all testing tuples. The results are shown in Figure 8 and Figure 9.

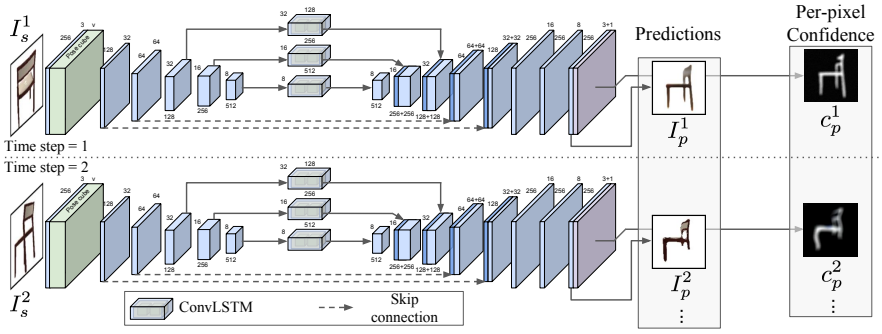

Fig. 2. The detailed architecture of the recurrent pixel generator. We use ConvLSTM in our residual blocks and therefore there is a hidden state that is passed through time that is not drawn.

E The Effect of Source Image Ordering

The source images are randomly ordered during training and testing. The goal of this paper is to maximize performance from what is given, not to find the best observation strategy. In applications where an agent can actively sample viewpoints, it would be interesting to investigate the effectiveness of observation orderings. Therefore we conduct a simple experiment where we test the model on all possible order. We randomly sampled 1000 tuple of source (image, camera pose) pairs from ShapeNet cars and chairs, and evaluated on all 24 ordering. On average, we have found that feeding the best order increases the performance (L_1 loss) by 2.382%/6.250% (car/chair), while feeding the worst order of source images causes a 2.583%/6.958% drop. Although our model shows some robustness to ordering, it is left for future work in learning an observation strategy.

F How Our Model is Fed with Source and Target Poses

We represent discrete pose (object) as an 18 element vector indicating the azimuth angle (sampled in the range [0, 340] with 20-degree increments) and a 3 element vector indicating the elevation (0, 10, or 20). We represent continuous pose (scene) as a continuous 6DoF vector specifying translations and rotations. We feed the pose to the network, by computing the difference between the source pose and the target pose $p_{diff} \in \mathbb{R}^v$ by $p_{target} - p_s$, where v denotes the dimension of the pose vector (21 for an object and 6 for a scene). We then tile p_{diff} along the spatial dimensions to obtain an input pose tensor $p_{input} \in \mathbb{R}^{H \times W \times v}$. Finally, the pose tensor is concatenated to the source image along the channel dimension, resulting in an input tensor $T \in \mathbb{R}^{H \times W \times (v+3)}$.

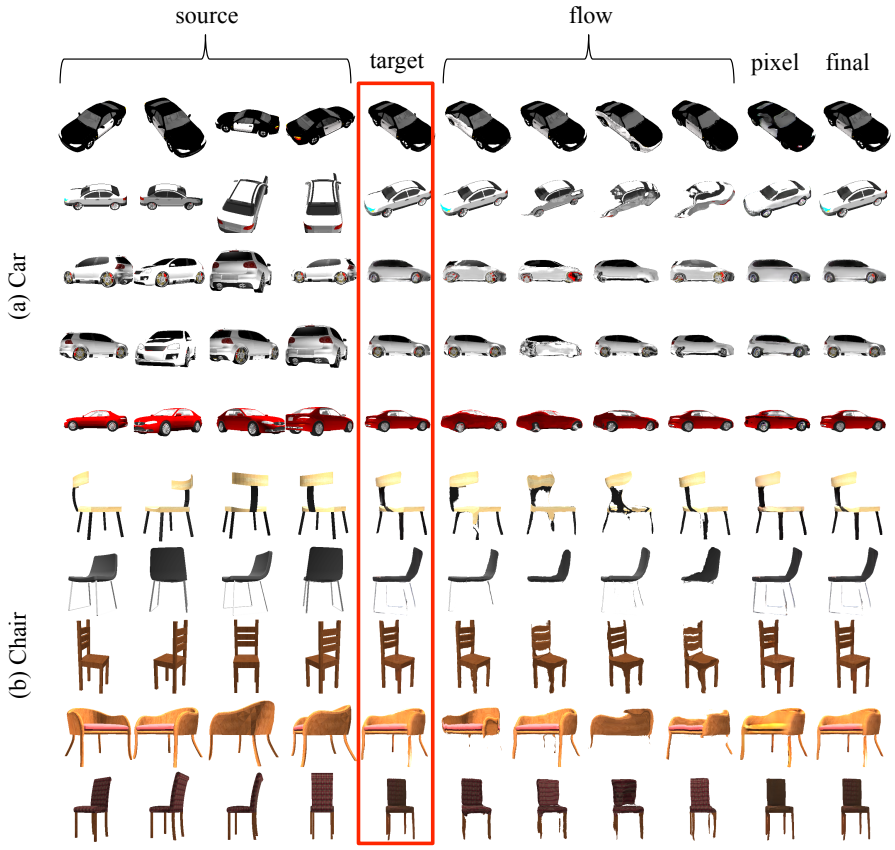

Fig. 3. Additional results on ShapeNet [5]. Each row presents testing tuples (source and target images), the intermediate predictions made by our proposed flow and pixel module, and an aggregated image. The pixel module synthesizes well-structured but sometimes less sharp or inaccurate colored (the second last chair) predictions, while the flow module produces visually realistic but sometimes incomplete objects (the tires of the cars and the legs of the chairs). Aggregate images, incorporating the strengths of both the two modules, are structurally coherent and visually appealing.

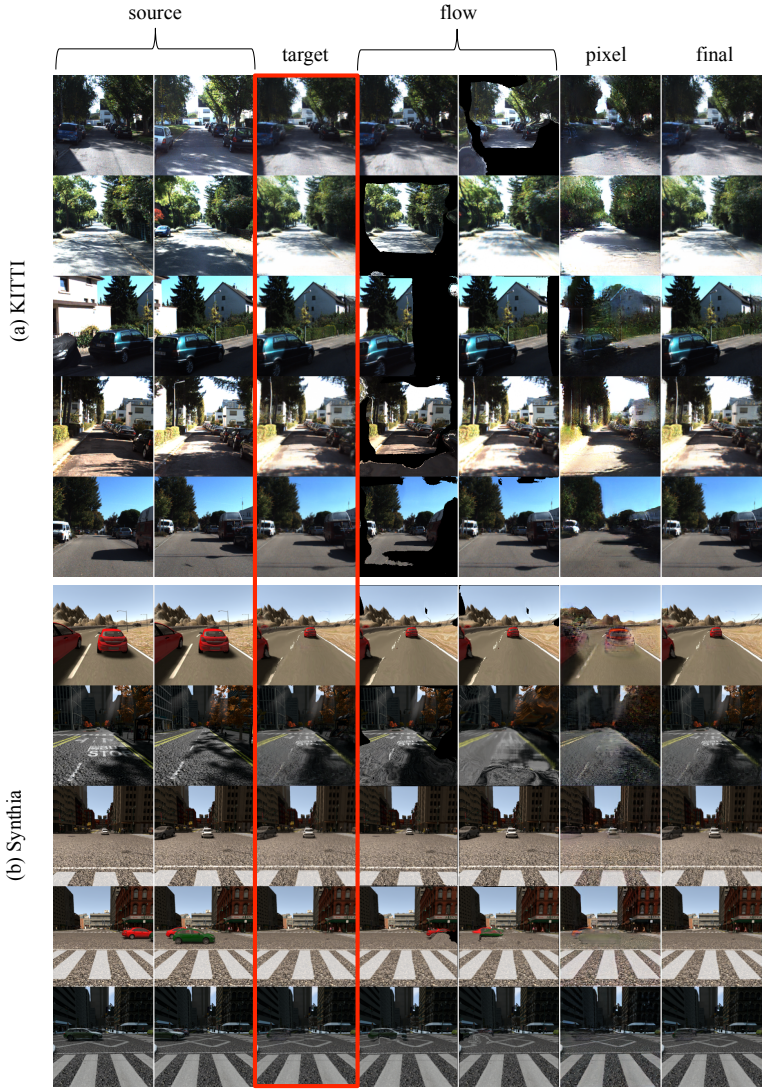

Fig. 4. Additional results on KITTI [6] and Synthia [7]. Each row presents testing tuples (source and target images), the intermediate predictions made by our proposed flow and pixel module, and an aggregated image. The pixel module synthesizes well-structured but sometimes less sharp or inaccurate colored (the car presented in the third row of KITTI) predictions, while the flow module produces visually realistic but sometimes incomplete objects (blank areas filled with black). Aggregate images, incorporating the strengths of both the two modules, are complete and visually appealing.

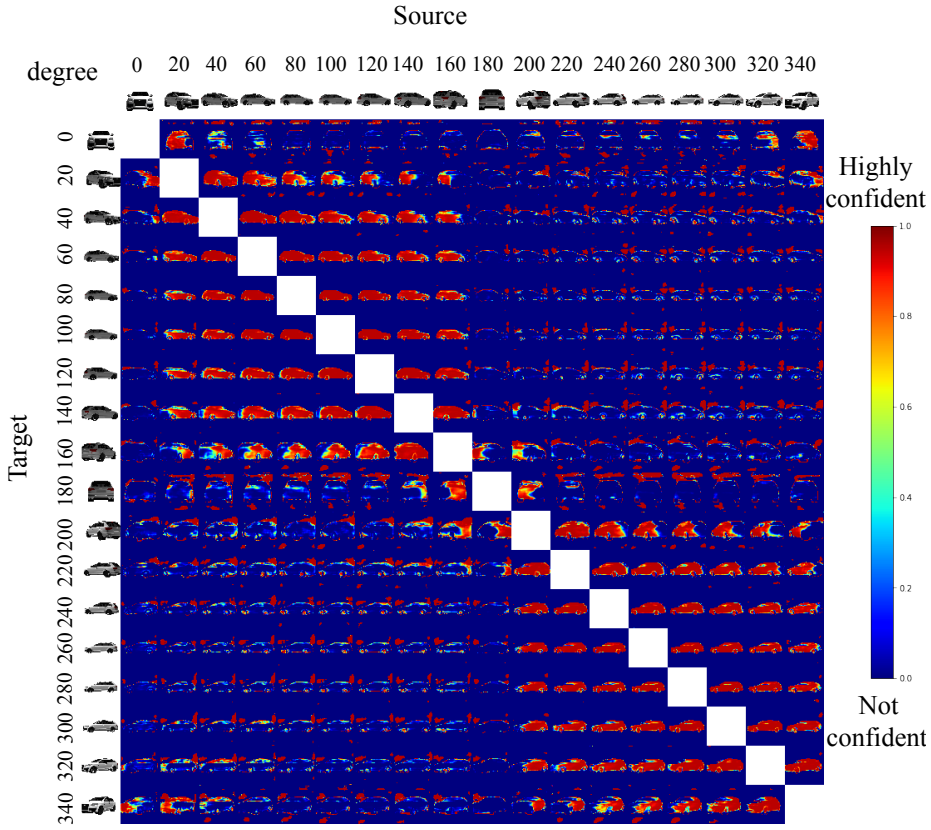

Fig. 5. Visualization of the predicted confidence maps on a car model. Each entry represents the predicted confidence map for a given source image and target pose. The confidence is represented using the jet-colormap, where red indicates highly confident, and blue indicating the otherwise.

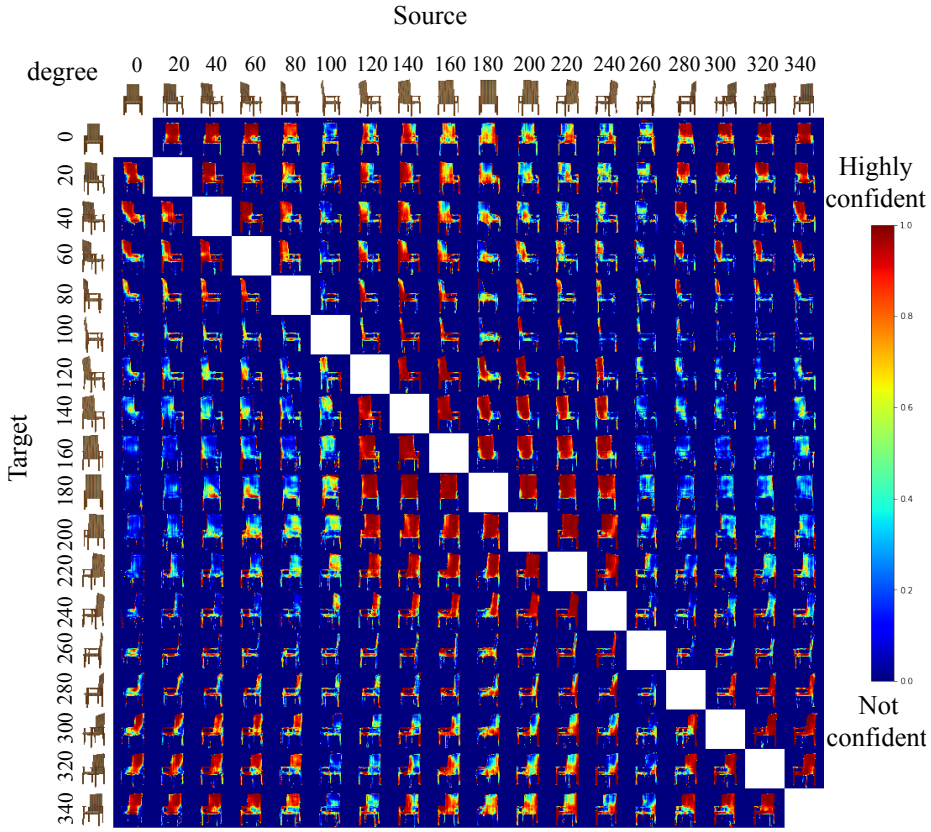

Fig. 6. Visualization of the predicted confidence maps on a chair model. Each entry represents the predicted confidence map for a given source image and target pose. The confidence is represented using the jet-colormap, where red indicates highly confident, and blue indicating the otherwise.

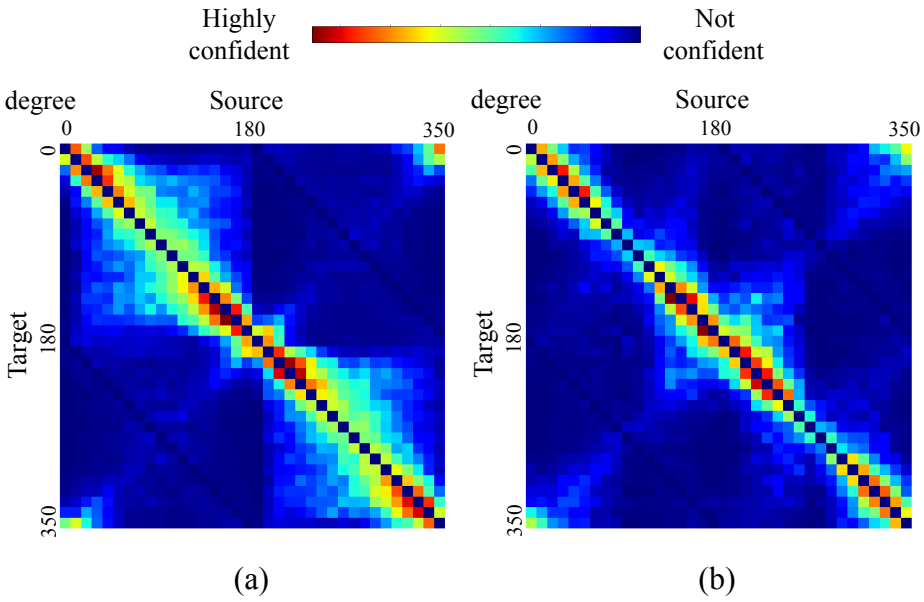

Fig. 7. Visualization of summation of the predicted confidence maps on all car (a) and chair (b) models. Each grid represents the summation of all predicted confidence map with a given source pose and target pose. The confidence is represented using the jet-colormap, where red indicates highly confident, and blue indicating the otherwise.

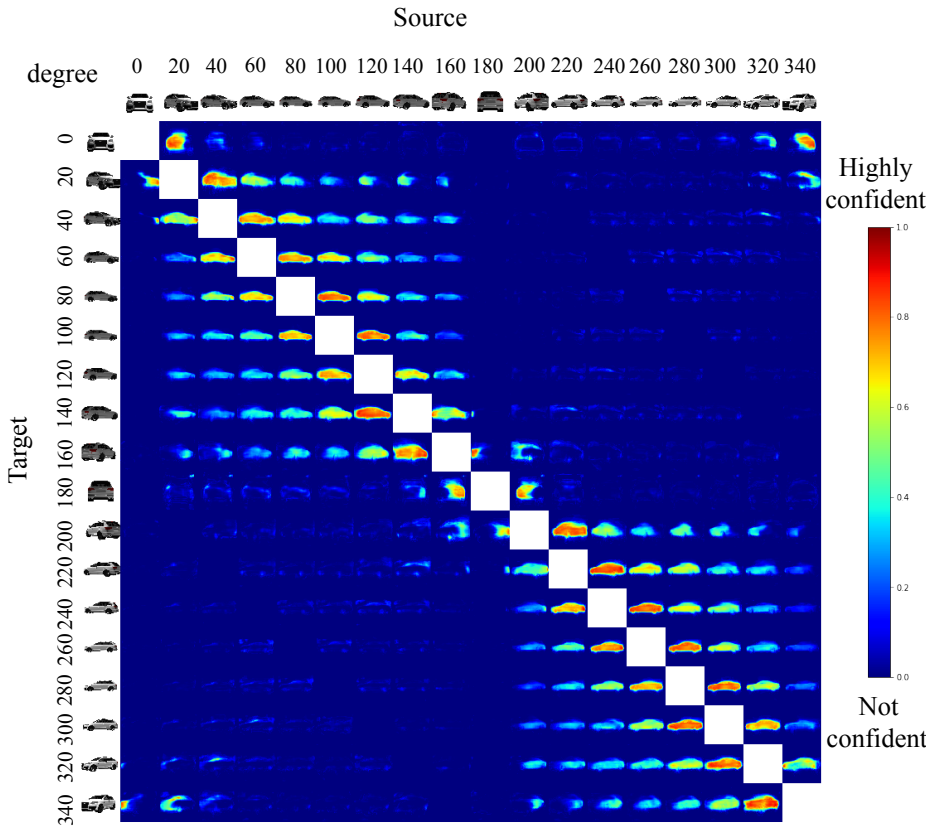

Fig. 8. Visualization of the *averaged* predicted confidence maps on a car model. To show the general tendency of predicted confidence maps, each entry represents the *averaged* predicted confidence map for a given source and target pose. Note that each entry is not computed from only a pair of source and target images but all the testing tuples with this source and target pose. In other words, each averaged confidence map is obtained by *averaging across confidence maps predicted for different car models*. The confidence is represented using the jet-colormap, where red indicates highly confident, and blue indicating less confident.

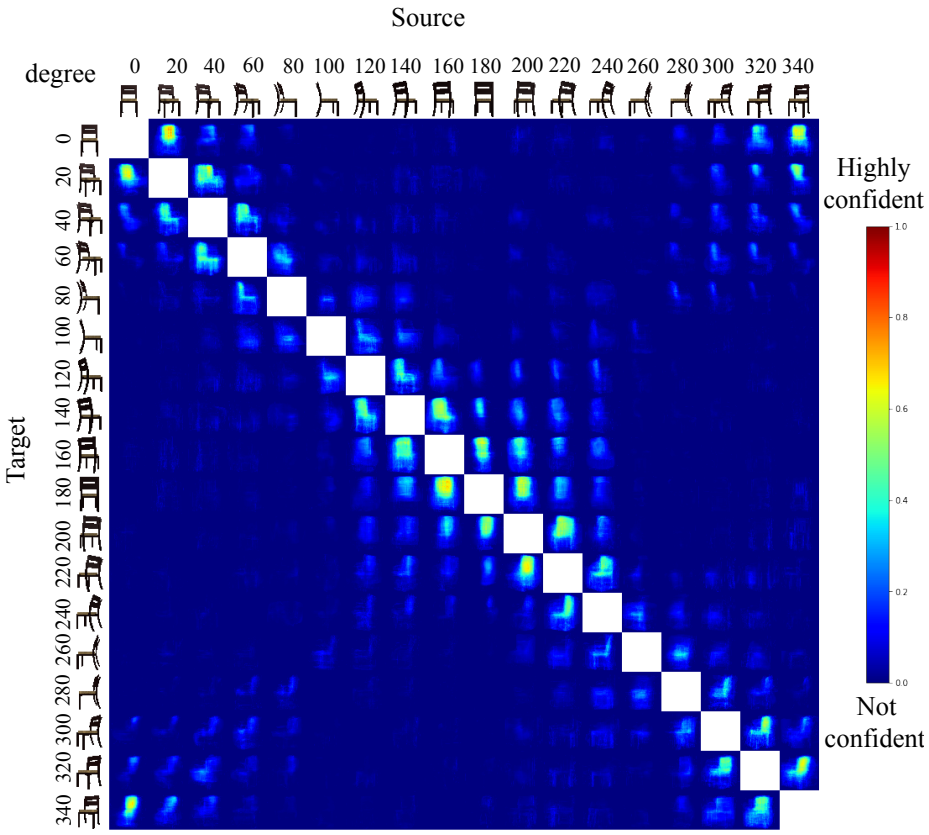

Fig. 9. Visualization of the *averaged* predicted confidence maps on a chair model. To show the general tendency of predicted confidence maps, each entry represents the *averaged* predicted confidence map for a given source and target pose. Note that each entry is not computed from only a pair of source and target images but all the testing tuples with this source and target pose. In other words, each averaged confidence map is obtained by *averaging across confidence maps predicted for different chairs models*. The confidence is represented using the jet-colormap, where red indicates highly confident, and blue indicating less confident.

References

1. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems. (2015) 802–810 1
2. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016) 1
3. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016) 1
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 1
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. Technical Report arXiv:1512.03012 [cs.GR] (2015) 4
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 5
7. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Computer Vision and Pattern Recognition (CVPR). (2016) 5