Supplementary Material
3D-RelNet: Joint Object and Relational Network for 3D Prediction

Nilesh Kulkarni[1] Ishan Misra [1] Shubham Tulsiani[2] Abhinav Gupta[1]
[1]Carnegie Mellon University [2]University of California, Berkeley
https://nileshkulkarni.github.io/relative3d/

A. Appendix

A.1. Metrics

We use the metrics from [3] and summarize them below.

- Translation ($\mathbf{t}$): Euclidean distance between prediction and ground-truth $\|\mathbf{t}_{\text{p}} - \mathbf{t}_{\text{gt}}\|$. $\delta_{\mathbf{t}} \leq 0.5$m.
- Scale ($\mathbf{s}$): We measure the average unsigned difference in log-scale, *i.e.*, $\Delta(\mathbf{s}_{\text{p}}, \mathbf{s}_{\text{gt}}) = \frac{1}{3}\sum_{i=1}^{3} |\log_2(\mathbf{s}^i_{\text{p}}) - \log_2(\mathbf{s}^i_{\text{gt}})|$. We threshold at $\delta_{\mathbf{s}} \leq 0.2$.
- Rotation ($\mathbf{q}$): Geodesic distance between rotations $\frac{1}{\sqrt{2}}\|\log(R_{\text{p}}^{\intercal} R_{\text{gt}})\|$. $\delta_{\mathbf{q}} \leq 30^{\circ}$. For objects that exhibit rotational symmetry, we use the lowest error across the different possible values of R_{gt}.
- Shape ($\mathbf{V}$): Following [1], we measure the intersection over union (IoU) and use threshold $\delta_V = 0.25$. As a higher IOU is better, so we use $\delta_V \geq 0.25$ for true positive.
- Bounding Box overlap ($\mathbf{b}$): The bounding box overlap is measured using IOU. $\delta_b \geq 0.5$.
- Detection: A prediction is considered a true positive when it satisfies the thresholds for each of the above components $(\delta_{\mathbf{t}}, \delta_{\mathbf{s}}, \delta_{\mathbf{q}}, \delta_V, \delta_{\mathbf{b}})$. We use Average Precision (AP) to measure the final detection performance.

A.2. Training Details

Unary Loss Functions. We use the following loss functions to train the unary predictors

- Loss Translation. $L_t = \|t_p - t_{gt}\|^2$
- Loss Scale. $L_s = \|\log(s_p) - \log(s_{gt})\|^2$
- Loss Rotation. $L_r = -\log(q_{gt})$, we minimize the NLL of the gt bin. q represents a probability distribution over the 24 bins.
- Loss Shape. $L_v = \sum_n V_n \log(\hat{V}_n) + (1-V_n)\log(1-\hat{V}_n)$. V_n are the ground-truth voxels, and $\hat{V}_n$ are the predicted voxels

Relative Loss Functions. We use the following loss functions to train the relative predictors

- Loss Translation. $L_{rt} = \|t_{(ij),p} - t_{(ij),gt}\|^2$, for objects $t_{(ij)}$ represents relative translation between i,j.
- Loss Scale. $L_{rs} = \|\log(s_{(ij),p}) - \log(s_{(ij),gt})\|^2$, for objects $s_{(ij)}$ represents relative scale between i,j.
- Loss Direction. $L_d = -\log(d_{gt})$, we minimize the NLL of the gt bin. d represents a probability distribution over the relative directions.

Joint Relative Losses We impose a loss on the joint prediction by combining unary and relative predictions

- Loss Translation. $L_{jt} = \|t^* - t_{gt}\|^2$, where t^* is the joint prediction computed using Equation 1
- Loss Scale. $L_{js} = \|\log(s^*) - \log(s_{gt})\|^2$, where s^* is the joint prediction computed using Equation 1

Optimization. We train our network in two stages. In the first stage of training we use ground truth boxes. We train for 8 epochs by using adam optimizer with a learning rate of 10^{-4}. During the first 4 epochs of the training we train for relative and object specific predictions independently and during next 4 epochs of the training we optimize the whole model jointly by combining the relative and object specific estimates. In the next stage we consider overlapping proposals with IOU of over 0.7 with respect to ground truth boxes and the ground truth boxes as positive proposals to further make the model robust in the detection setting.
In the NYUv2 setting we start with a network trained on the SUNCG dataset and finetune the network for 16 epochs on the NYU train + val split and evaluate method on the test split.

Rotation Prediction. We defined $\Delta(R, d, t)$ as a measure of how inconsistent a predicted rotation R is w.r.t the predicted relative direction distribution d and relative translation t. Given a predicted rotation R, we would expect the

predicted direction to align with the vector $\bar{R}\ \hat{t}$, where $\hat{t}$ is unit-normalized. Note that the predicted d is a probability distribution over possible directions, and let d^* denote the bin that aligns maximally with $\bar{R}\ \hat{t}$. We measure $\Delta(R, d, t)$ by combining measures of how likely this bin is with how well it agrees with the rotation and translation: $\Delta(R, d, t) = -\log\ p(d^*) + (1 - \cos(d^*, \bar{R}\ \hat{t}\ ))$.

Relative Importance. We use lambda for unary importance to get $\mathbf{t}^*$ and $\mathbf{s}^*$ as 1. In case of rotation we use we weight for the relative predictions as $\min(5.0/n, 1)$ where n represents number of neighbours of the object. In the detection setting we create set of valid objects which are allowed to influence the final predictions for other objects based upon the detection score. We consider objects with a score above 0.3 to be part of the valid set and only use them to get final predictions for other objects.

A.3. Additional Visualizations and Results

We visualize the precision-recall curves in the detection setting using the SUNCG dataset in figure 1. We also visualize predictions for randomly sampled images in the setting with known bounding boxes in figure 4.

A.4. Visualization on NYU in Detection Setting

We visualize sample from NYU in the detection setting, and show comparisons with respect to the baseline. Please refer to Figure 2.

A.5. Factored3D + CRF Details

We implement the CRF model by creating statistical models for relative translation, relative scale, and relative direction for every pair of object categories. We fit a mixture of 10 Gaussian to the data from each pair and modality. At test time we optimize using this prior assuming access to ground truth class labels to choose the appropriate prior. For optimization we use LBFGS [2] and we optimize for 1000 iterations for every example. We visualize the outputs for CRF + Factored3D model and compare against the baseline in Figure 3

References

[1] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1

[2] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. 2

[3] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018. 1

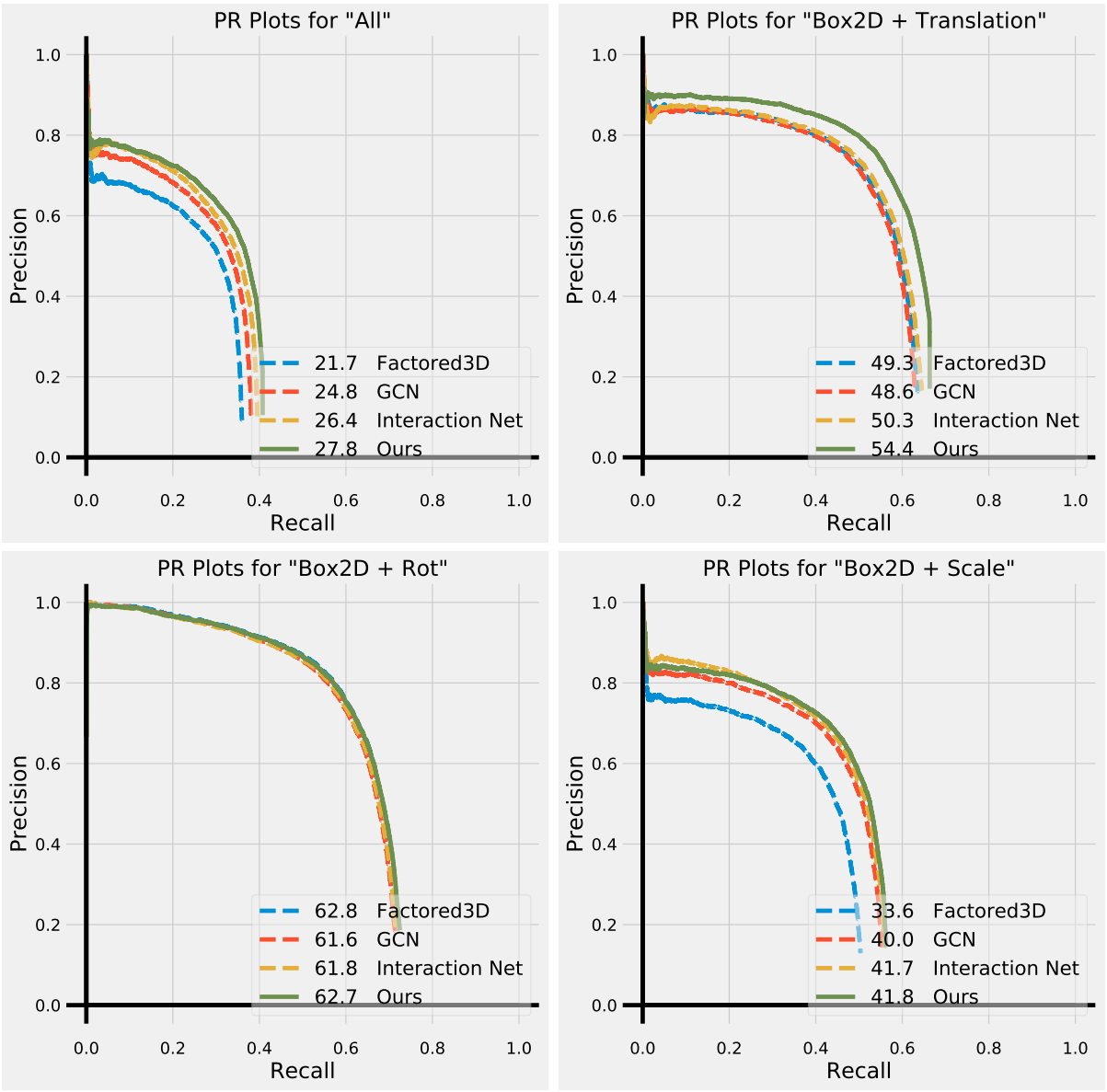

Figure 1: We plot the precision-recall (PR) curves for the **detection setting** for SUNCG and also display the mean Average Precision (AP) values in the legend. In each of these curves, we vary the criteria used to determine a true positive. This helps us analyze the relative contribution of each component (translation, rotation, scale) to the final performance.

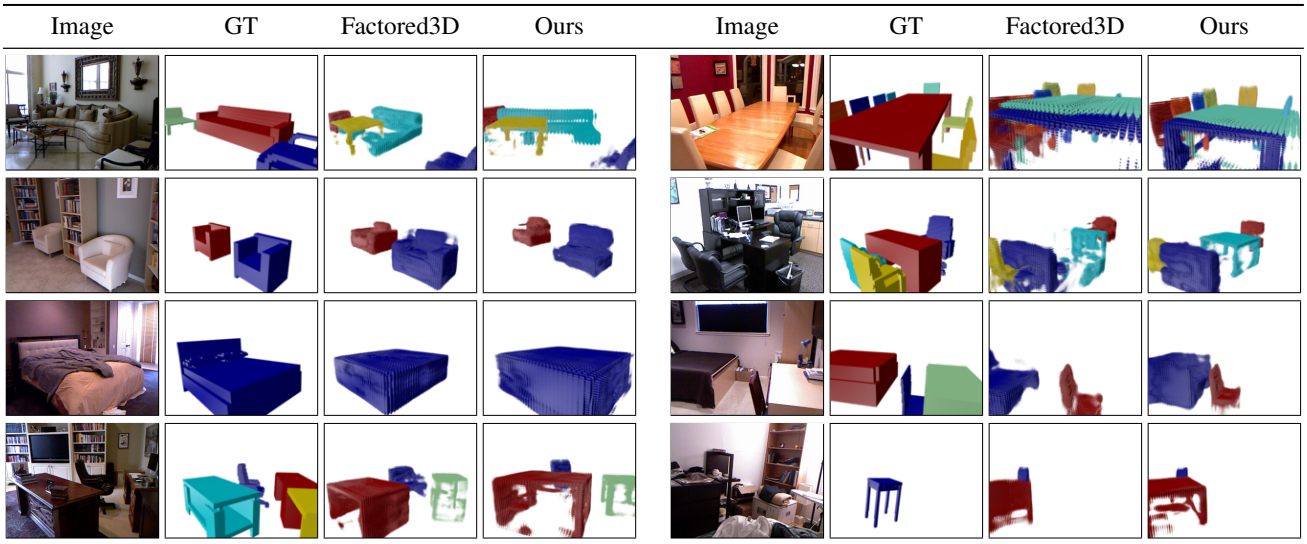

Figure 2: Detection Setting for NYU: We visualize sample prediction results in the detection setting Section 4.3 of the main manuscript. We can notice that relative arrangement between objects is better under the Ours column vs Factored3D.

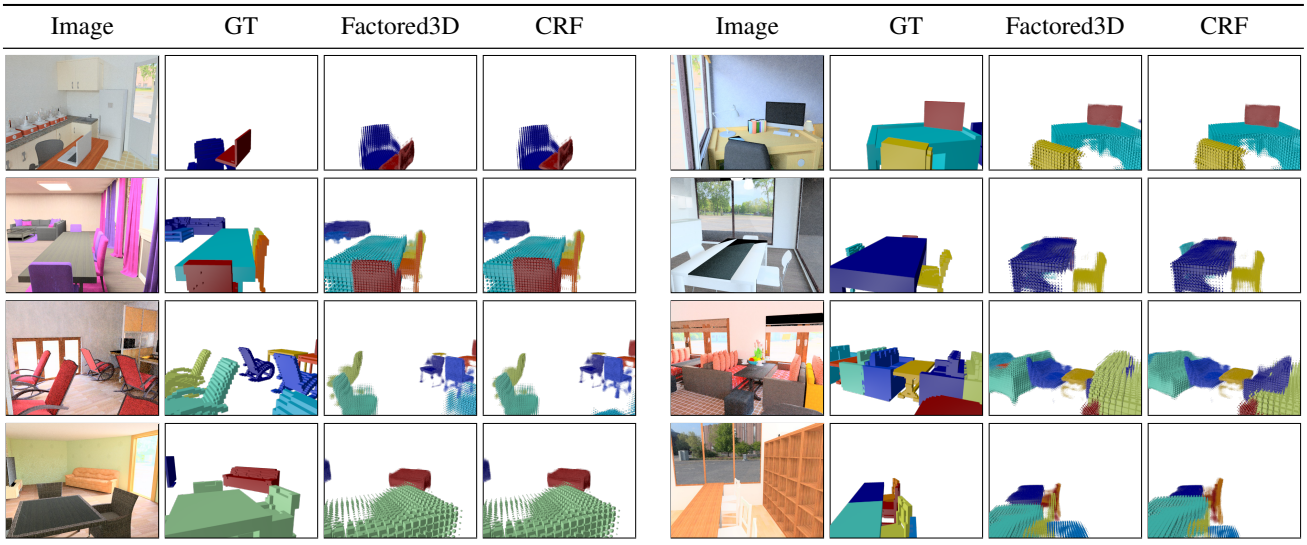

Figure 3: Factored3D + CRF in GT Box setting: We visualize sample prediction results in the GT Box setting Section 4.2 of the main manuscript for the Factored3D + CRF model. The first two rows show examples where the Factored3D + CRF model does better than Factored3D baseline while the next two rows show the examples where it does worse.

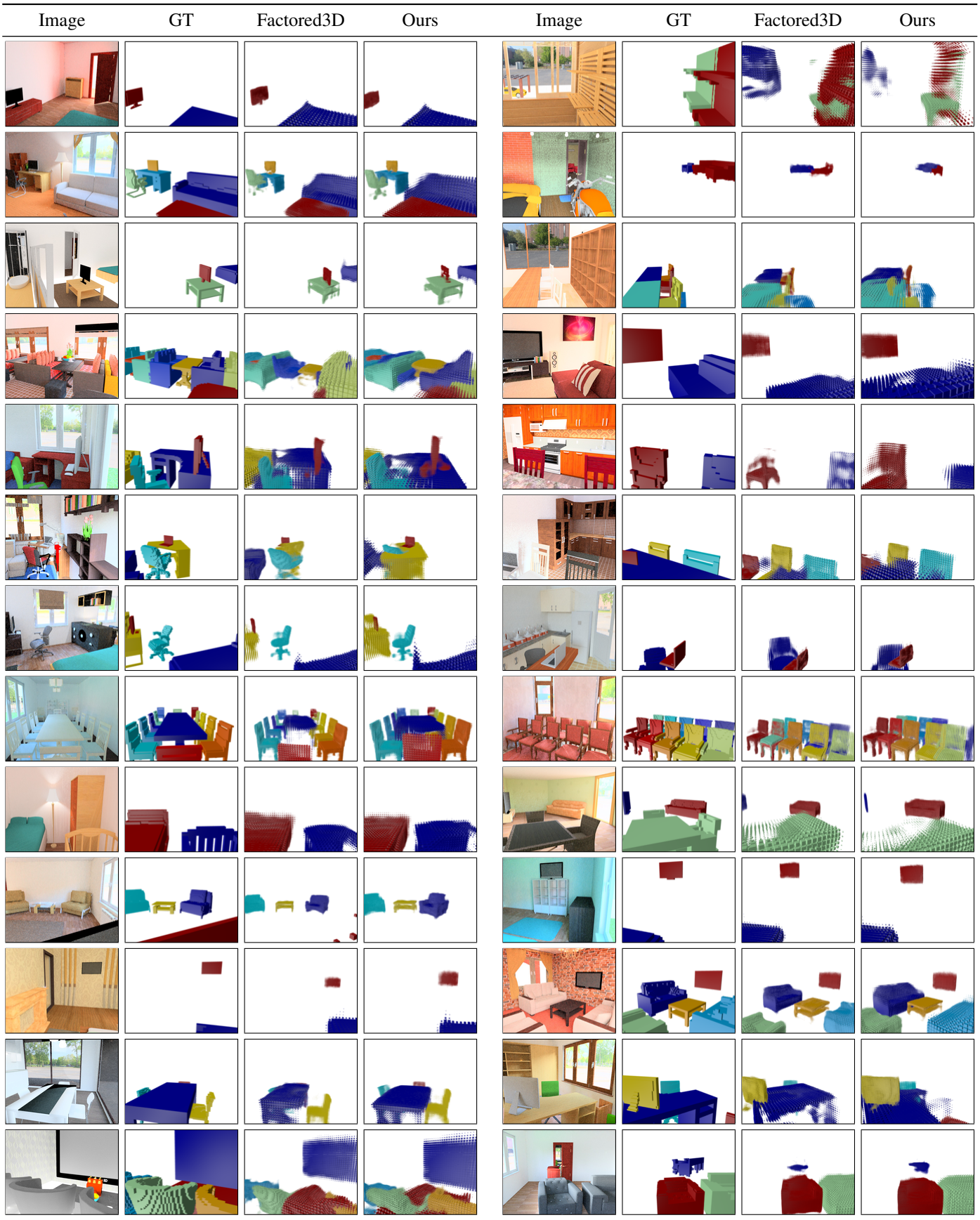

Figure 4: We visualize predictions for *randomly sampled images* in the setting with known ground-truth boxes for the SUNCG dataset.