Supplementary Material

Nitin Bansal, Pan Ji, Junsong Yuan, Yi Xu

AffineMix Data Augmentation In this subsection we illustrate the steps followed to generate augmented images using the proposed AffineMix method. We call AffineMix, an intra data augmentation technique, as we do not use any other image to create the augmented image, but use the same target frame I and then apply random affine transformation on a selected movable object to get a new image I' as shown in algorithm 1.

Algorithm 1 Intra Data Augmentation

 # Intra Data Augmentation step:
$Scale \leftarrow random(0.5, 1.5)$
$t_x = (1.0 - 1/scale)) * (o_x)$ ▷ Offset along x
$t_y = (1.0 - 1/scale)) * (o_y)$ ▷ Offset along y

 # Affine transform according to Scale:
$I_a = aff\_transform(I, 1/scale, [t_x, t_y])$
$L_a = aff\_transform(L, 1/scale, [t_x, t_y])$
$S_a = aff\_transform(S, 1/scale, [t_x, t_y])$
$D_a = aff\_transform(D, 1/scale, [t_x, t_y])$
$D_a = Scale * D\_a$

 # Generate new image
$M = D_a \leq D$ ▷ foreground mask
$I' = M \odot I_a + (1 - M) \odot I$ ▷ Augmented Image
$L' = M \odot L_a + (1 - M) \odot I$ ▷ Augmented Label
$S' = M \odot L_a + (1 - M) \odot S$ ▷ Augmented Softmax

We choose a random scale for depth which lies in range(0.5, 1.5), and then perform affine transform on the selected moving object mask M to achieve the new augmented frames, which are then used during semi-supervised semantic segmentation learning. Fig. 1 provides further qualitative examples of proposed data augmentation scheme.

Orthogonality In our ablation study, we confirm that enforcing orthogonality has a positive impact on both depth and semantics performance. Here we show its effect on features of different layers of the depth and semantics decoder module, once the training has ended. Suppose a layer of depth decoder has a dimensionality of (BxCxHxW), where B,C,H and W represents batch size, number of channels, height and width respectively. We estimate an average inter-channel correlation, according to following algorithm:

Figure 1. **Example Images AffineMix**:Left-to-right: Original Image, GT Labels, Scaled foreground mask, Augmented Image, New label space

Algorithm 2 Inter Channel Correlation

Require: $layer$ ▷ Intermediate CNN feature layer
$B, C, H, W = layer.dim$
while $c \leq C$ **do**
  $norm = 0.0$
  while $i \leq C$ **do**
    if i = c **then**
      $continue$
    else
      $corr = layer[:, i, :, :] * layer[:, c, :, :]^T$
      $norm+ = l1\_norm(corr)$
    end if
  end while
  $norm = norm/C$ ▷ average norm per layer
end while

We calculate the average norm as specified in Algo. 2, for all decoder layers for both the tasks, with and without OR(orthogonal regularization). In Fig. 2 (ii), we plot average difference of correlation for four different layers of semantics and depth decoder module, without and with OR. As seen in the plot, we observe a positive difference all along for both semantics and depth module, suggesting greater inter-channel independence.

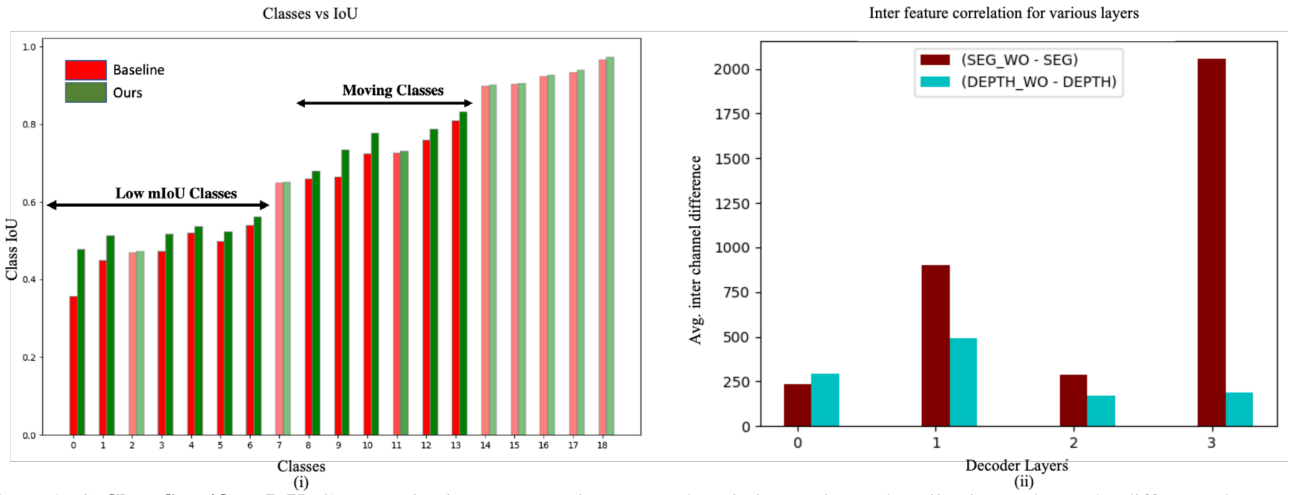

Figure 2. (i) **Class Specific mIoU:** Comparative improvement in IoU numbers in increasing order. (ii) Figure shows the difference between the average correlation between all features for semantics and depth decoder, for without (WO) OR and with OR.

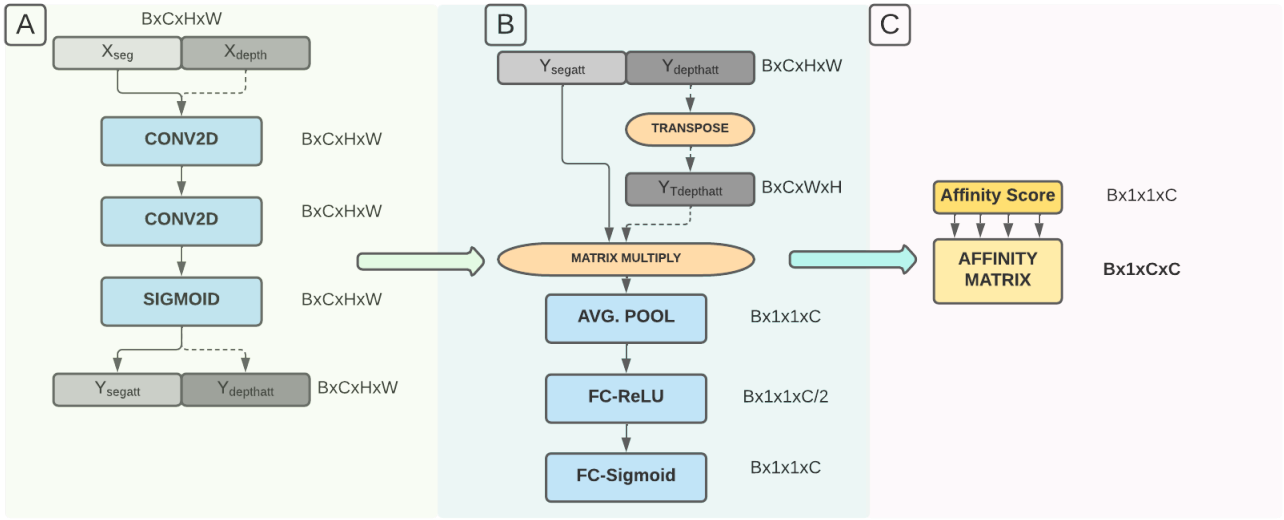

Figure 3. Building blocks of the **Cross Channel Affinity Module**.

CCAM Module As seen in Fig. 3, architecture of CCAM module can be divided in to three blocks namely **A**, **B** and **C**, which serves three different purpose. Block A acts as a spatial attention layer, where as Block B estimates cross feature correlation and follows it up with channel attention layer to give us an *affinity score* per feature channel, which is then followed by Block C, which does a simple channel-wise accumulation of *affinity score* to give final Affinity Matrix. Fig. 3 shows individual learnable layers part of all the three blocks in complete detail. We also present the steps followed for calculating affinity matrix and getting resultant depth and semantic features in algorithm 3.

Further Qualitative Results We have discussed in the main paper there are certain classes within the 19 categories present in the cityscapes dataset which have shown more improvement compared to other *saturated classes*. We have marked them as *low-mIoU-classes* and *movable-classes* which were mainly ("rider", "motorcycle", "wall", "bus", "truck", and "train") respectively. For this we plot mIoU numbers in an increasing order [1] As seen in Fig. 2(i), we see that most of the improvement is seen for classes belonging to *low-mIoU-classes* and *movable-classes*. We particularly highlight the some of the positive examples of

[1] Order: motorcycle, wall, fence, rider, pole, traffic-light, terrain, traffic-sign, bicycle, train, truck, person, sidewalk, bus, building, vegetation, car, sky, road.

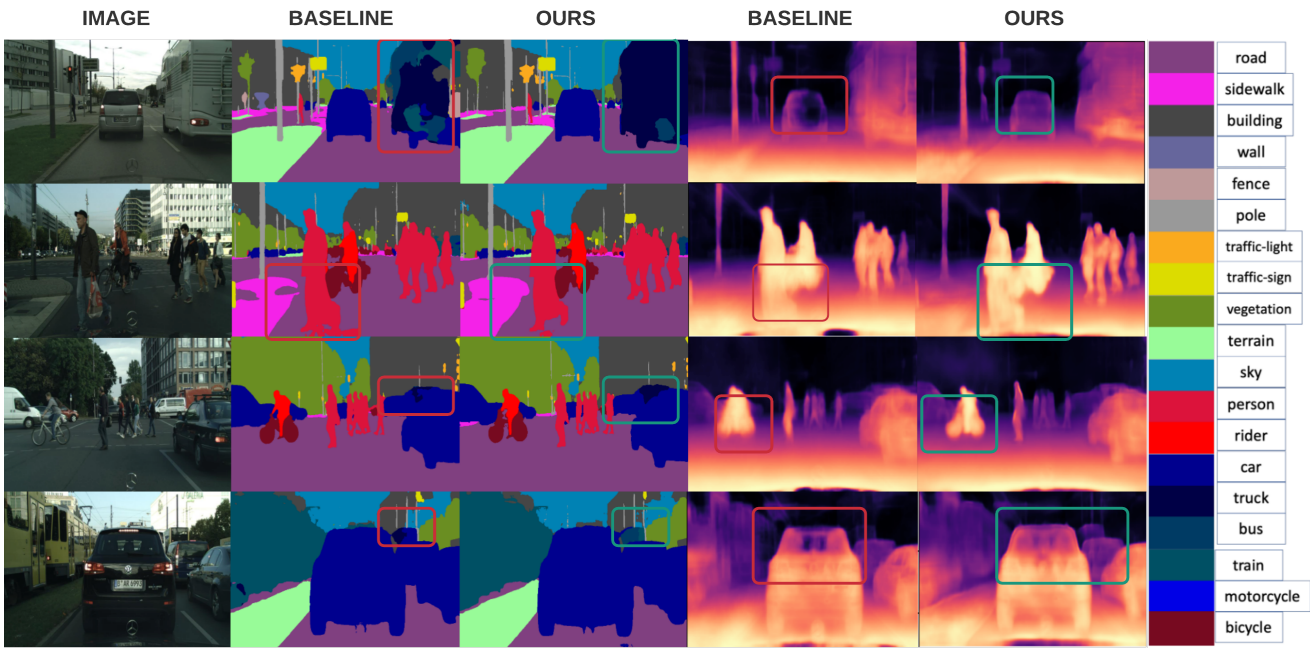

Figure 4. **Example Outputs:** Semantics and depth results for the same input images, showing comparative results with the baseline.

Algorithm 3 Cross Channel Affinity Block

Require: X_{seg}, X_{depth} ▷ Semantics and Depth features

 # Estimate cross channel affinity:
$Y_{segatt} = Spatial\_Attn(X_{seg})$
$Y_{depthatt} = Spatial\_Attn(X_{depth})$
$Y_{Tdepthattn} = Transpose(Y_{depthatt})$
$i \leftarrow 0$
while $i < C$ **do** ▷ C represents number of channels
  $\alpha_i = Channel\_Affinity(Y_{segatt} * Y_{Tdepthattn})$
  $C_T = C_T + \alpha_i$ ▷ concat across dimension
end while

 # Mutual features sharing between tasks:
$i \leftarrow 0$
while $i < C$ **do**
  $X_{seg} = X_{seg} + X_{depth_i} * C_{Ti}$ ▷ along row dimension
end while
$j \leftarrow 0$
while $j < C$ **do**
  $X_{depth} = X_{depth} + X_{seg_j} * C_{Tj}$ ▷ along column dimension
end while

these classes, as seen across different cities for test set of cityscapes dataset in Fig. 5. We find there are quite a few examples, where the mistakes are mainly due to a mix-up in the predictions mainly concerning the labels belonging to class set of bus, car, truck and train due to obvious visual similarity. In Fig. 4, we show qualitative depth and semantics results for the same set of input images, highlighting improvement obtained on both tasks. Here, it is imperative to state that it's not necessary that we see an improvement for both the tasks, over all input test images. There are few classes which show minimal improvement and also certain degenerate example images where we fail to effectively handle *unknown* classes, examples of which are discussed in Sec. 1. Example Qualitative result for ScanNet data can be seen in Fig. 7

Training Details: Our Proposed model achieves better performance with approximately the same number of parameters (>2000 more parameters than baseline). There is a slight increase in the number of flops as seen in Tab. 1, which can be mainly attributed to the matrix multiplication step, used during the correlation calculation step in Block *B* (See Fig. 3).

More Ablation Experiments We ran further set of ablation experiments to verify the efficacy and impact of each individual component. Table 2 presents the result for all conducted experiments in detail. Observations are consistent with that seen in the main paper. Inclusion of CA helps in further consolidating the gain for depth estimation achieving the best absolute relative error of **0.140**, whereas adding OR and AM module helps in improving the base result by **0.38** and **0.39** mIoU numbers respectively. We do

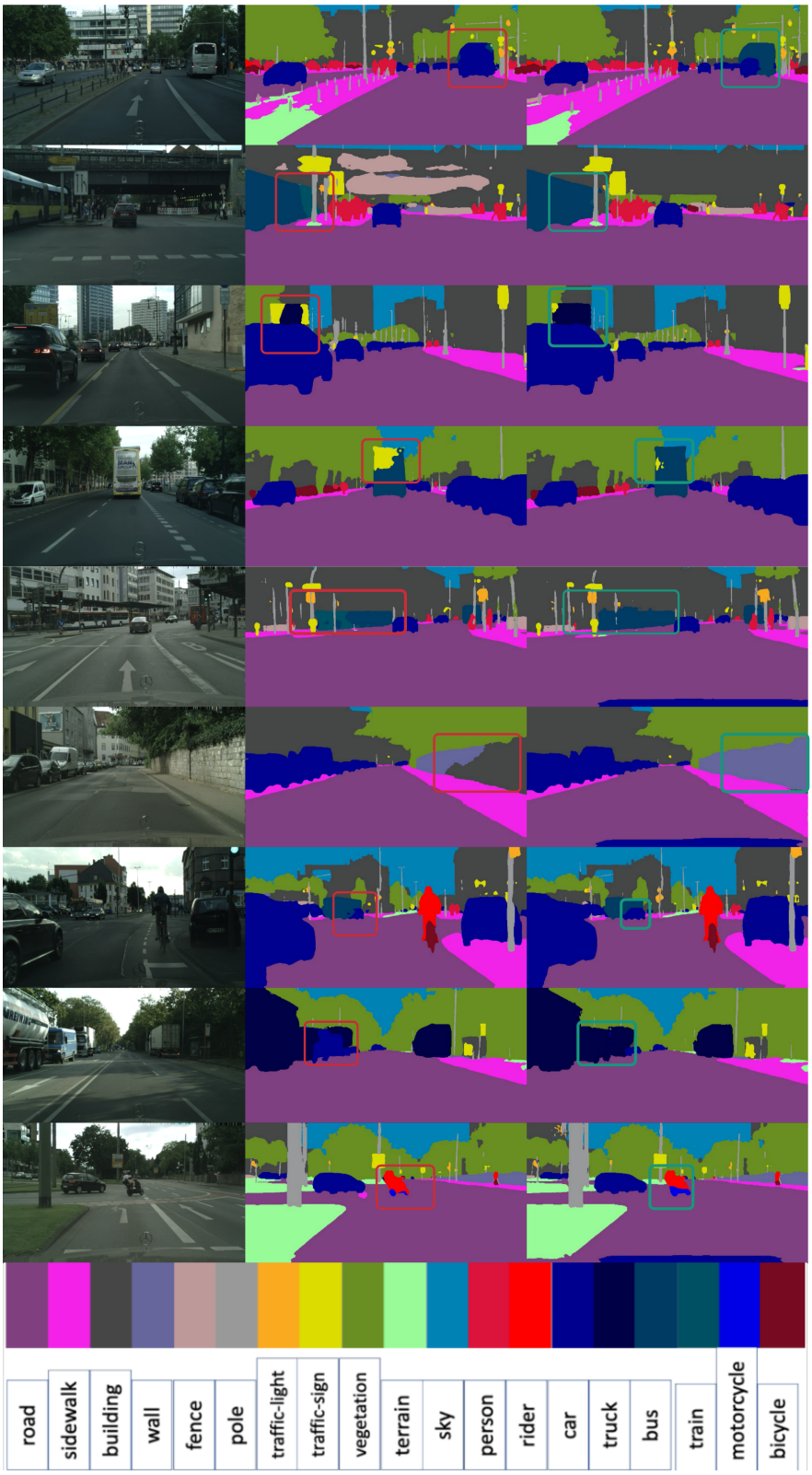

Figure 5. **Examples of Improved classes:** From up-to-down:(Bus-Car), (Bus-Train), (Truck-Sign), (Truck-Sign), (Bus-Train), (Wall-Building), (Bus-Car),(Truck,Car), (Rider, Motorcycle)

Model	Flops^{+} (10^9)	Params (10^6)
Baseline	-	87.226
Ours	0.27	87.228

Table 1. Comparative Training details. + denotes additional numbers of flops used.

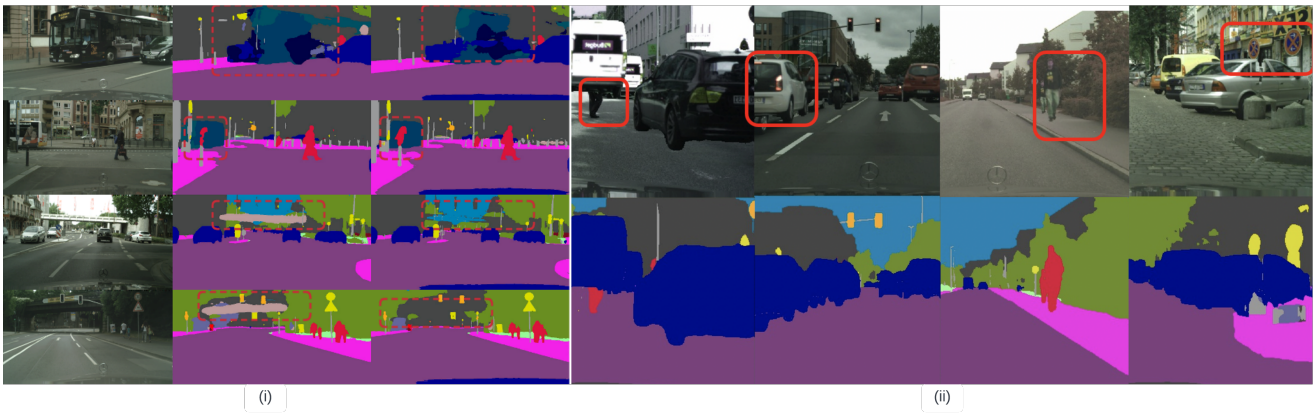

Figure 6. **Bad Examples:**(i) Few examples where the model gets confused and makes the wrong prediction. Left-to-Right: Input image, Baseline, Ours.

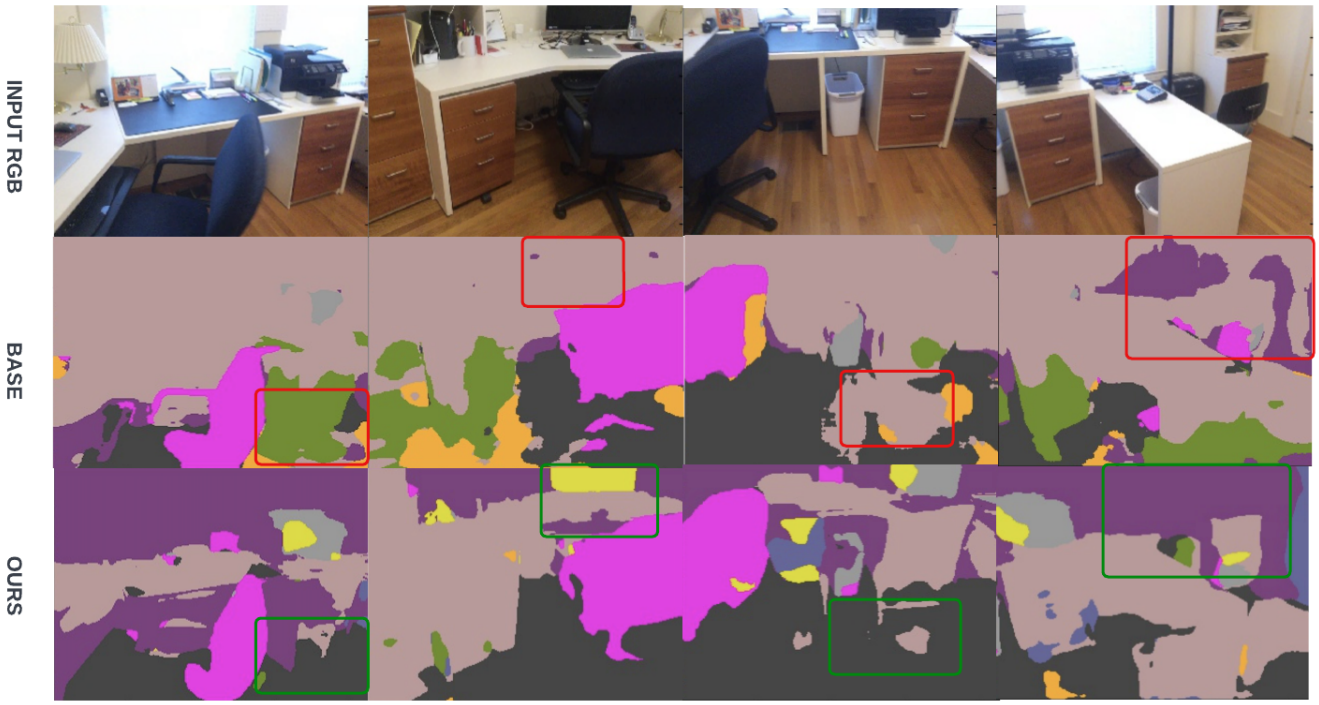

Figure 7. Example improvement between Base and Our model for scene0166_00 in ScanNet dataset. Above image shows improvement for 'Floor', 'Screen', 'Floor' and 'Wall' class respectively.

see a slight dip in mIoU while using CA, but when combined with AM/OR it always performs better even for semantic segmentation.

Quantitative Results We present class wise mIoU number for both ScanNet and Cityscapes dataset as shown in Table 3 and 4. For ScanNet, we see improvement for all the classes except for class *Chair* and *Floor*, Where as for Cityscapes, we see improvement across all the classes, albeit to varying degree, with *low-mIoU-classes* and *movable-classes* seeing the majority of gain.

Dataset: Cityscapes Cityscapes [2] consist of 2,975 training and 500 validation images with ground-truth se-

Model	Seg. Metrics	Depth Metrics					
	mIoU↑	AbsRel↓	SqRel↓	RMSE↓	a1↑	a2↑	a3↑
Ours (CCAM)	69.35	0.142	1.653	7.230	**0.824**	**0.957**	0.988
Ours + OR	69.73	0.143	1.612	7.433	0.817	0.954	0.987
Ours + CA	69.13	**0.140**	**1.638**	7.317	0.819	0.954	**0.988**
Ours + CA + AM	69.74	0.144	1.471	**7.158**	0.812	0.955	0.988
Ours + CA + OR	**69.77**	0.148	1.666	7.309	0.789	0.953	0.987

Table 2. Ablation experiments showing comparative mIoU and depth results for Cityscape dataset. CA: Color Aug, OR: Orthogonal Regularization, AM: Affine Mix

Model	Wall	Chair	Floor	Door	Table	Box	Screen	Cabinet	mIoU
Base	71.04	**62.90**	**64.50**	10.60	46.50	8.70	22.00	26.20	39.50
Ours(CCAM)	**72.40**	60.10	62.00	**21.30**	**48.30**	**11.60**	**25.10**	**31.80**	**41.57**

Table 3. Table presents the comparative performance of 8 different classes in ScanNet dataset between the baseline and CCAM enabled model.

Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	mIou
Base	35.74	44.92	46.97	47.21	52.11	49.88	54.12	65.02	66.07	66.60	72.56	72.66	76.00	80.93	89.93	90.41	92.42	93.34	96.79	68.09
Ours	40.53	52.60	47.97	53.11	53.63	53.30	56.14	66.38	67.47	75.17	77.08	73.31	78.14	83.91	90.35	90.57	93.03	93.78	97.12	70.72
Change	4.79↑	7.68↑	1.00↑	5.91↑	1.52↑	3.42↑	2.02↑	1.36↑	1.40↑	8.57↑	4.52↑	0.65↑	2.14↑	2.98↑	0.42↑	0.16↑	0.61↑	0.44↑	0.33↑	2.63↑

Table 4. Table presents the comparative performance for all 19 classes in Cityscapes dataset. Order of class is: motorcycle, wall, fence, rider, pole, traffic-light, terrain, traffic-sign, bicycle, train, truck, person, sidewalk, bus, building, vegetation, car, sky, and road.

mantic segmentation labels, collected from 21 different European cities. For semi-supervised segmentation, we use only 2,975 labeled training images, which are randomly split into a labeled and an unlabeled subset in accordance with number of labelled images being used for training.

ScanNet During our experiment with ScanNet [3], we mainly focus on scenes which are from *Living room*, *Bedroom* and *Office*, which mainly consist of 8 (parent) classes as shown in fig. 8, to make joint-training bit smoother. Using which we get a new train/val/split of 76/15/11 scenes respectively. All other classes were set to *ignore class* during both training and evaluation. Also, since we have the ground-truth pose for ScanNet, we use it during depth prediction. We follow similar data preprocessing steps for ScanNet during training and evaluation as done for Cityscapes.

Model Architecture It comprises of a shared encoder network which is ResNet101 [4] with output stride as 16. We use two different decoder modules for semantics and depth respectively, which is a combination of Deeplabv3 [1] with a U-Net [5] decoder. ASPP Blocks [4] with dilation rates of 6, 12, and 18 are used for aggregating encoder features of different scales. U-Net [5] decoder has five upsampling blocks with skip connections, with output channel as 256, 256, 128, 128, and 64 respectively.

1. Limitations

In this section, we briefly discuss specific limitations of our method. We start with illustrating examples of input images, where we fail to handle *unseen* and *confusing* scenarios and make wrong predictions. We find there are specific cases, as shown in Fig. 6(i) (row 1, 2), where picture/painting and transparent glass on the *bus*'s exterior makes it difficult for the model to classify *bus* correctly. We also find that images containing flyover or overhead bridges (See Fig. 6(i) (row 3, 4)) are either misclassified or missed. During our experiment, we also find that there is little impact mIoU-wise on classes such as traffic-sign, fence, and vegetation. In Figure 6 (ii), we highlight certain examples of augmented images that need not necessarily obey the geometry of the scene and have hanging, perspective and truncated artifacts (due to occluding stationary object) respectively. These are particularly seen in cases that involve large depth changes during the data augmentation and is enhanced due to using an average camera intrinsics for all scenes. Another direction of improvement would be to further study effects arising from incorrect lighting arising due to proposed data augmentation, which needs to be investigated to understand its impact on both tasks. We look forward to work towards finding meaningful answers for the aforementioned scenarios in future work.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for

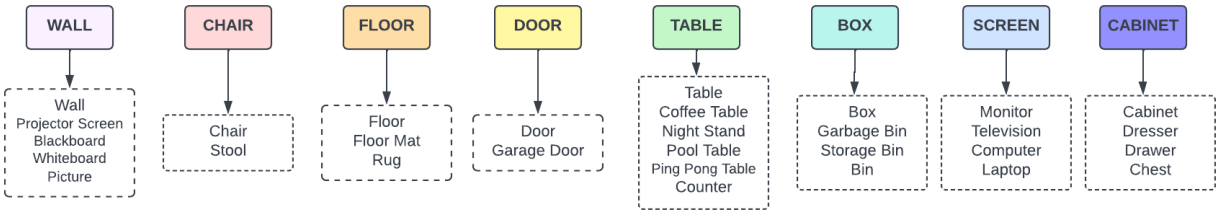

Figure 8. Above figure shows hierarchical class structure, with 8 parent classes. This also shows the mapping between classes defined in NYU [6] and ScanNet dataset.

semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[6] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.