Multi-domain semantic segmentation with overlapping labels Supplementary material

Petra Bevandić,* Marin Oršić,† Ivan Grubišić, Josip Šarić, Siniša Šegvić
University of Zagreb, Faculty of Electrical Engineering and Computing
name.surname@fer.hr, marin.orsic@gmail.com

This supplement presents additional validation experiments and offers further qualitative analysis of our submission to RVC 2020. Additionally, it provides visualizations of our training and evaluation mappings for the two baselines and our universal taxonomy.

We use Adam and attenuate the learning rate from $5 \cdot 10^{-4}$ to $6 \cdot 10^{-6}$ by cosine annealing. We use the largest batch size that fits into the GPU memory, and train our models for 100 epochs. All these experiments use single-scale evaluation.

1. Evaluation of the two baselines in multi-domain experiments

The main paper evaluates our universal taxonomy with a custom unified taxonomy called MSeg [3] in a multi-domain setup. We repeat that experiment with our two baselines, naive concatenation and partial merge, which in this setup have 469 and 307 classes. In order to have a consistent training setup, we train all four models on a single Tesla V100 32GB GPU. We use batch size 12, since that is the maximum for naive concatenation. We keep the rest of the training protocol the same as in the main paper.

The top section of Table 1 presents evaluation according to the original protocol. Our universal model prevails on most datasets. Partial merge performs comparably, since only 13 out of 307 classes overlap with some other class (our universal taxonomy has 294 classes). The bottom section of the table shows a similar outcome when we evaluate only on the 194 MSeg classes. We note that even naive union outperforms manual relabeling when original evaluation protocols are used since manual relabeling drops too many classes.

2. Validation experiments

We validate several hyper-parameters on the Vistas dataset. In all tables the columns correspond to training resolution (MPx), total batch size (BS), number of computational nodes used for training (nPU), and segmentation accuracy on Vistas val (mIoU).

*Equal contribution.

Evaluation protocol	Taxonomy	Ade20k	BDD	Cityscapes	COCO	IDD	SUN RGBD	Vistas
Original	naive concatenation	29.3	55.5	68.7	30.8	52.1	41.1	35.3
	partial merge	30	58.4	70.6	32.2	**54.4**	41.7	37.6
	MSeg	23.2	58.1	71.4	29	42.2	41.9	25.7
	Universal (ours)	**31.3**	56.5	71.2	**33.7**	53.1	**42.5**	37.9
MSeg	naive concatenation	32.3	55.5	68.7	31.5	53.9	41.1	41.4
	partial merge	33.3	58.4	70.6	32.9	51.8	41.7	43.9
	MSeg	34.3	58.1	71.4	33.5	53.9	41.9	43.0
	Universal (ours)	34.7	56.5	71.2	**34.6**	51.9	**42.5**	43.8

Table 1: Multi-domain experiments with SNp-RN18 on the seven MSeg datasets [3]. We train the baselines and our universal model on original labels, and compare with a NLL model trained on manually relabeled images according to the MSeg taxonomy [3]. Both models are evaluated on validation subsets of Ade20k, BDD, Cityscapes, Coco, IDD, SUN RGB-D and Vistas. We consider all unmapped logits as class void.

2.1. Validation of segmentation architectures

Table 2 shows performance of various semantic segmentation architectures on the validation subset of Vistas. We present best approaches from the literature and compare them with pyramidal SwiftNets with different backbones. All our models use checkpointed backbones and were trained on 1 Tesla V100 32GB GPU.

Model	MPx	BS	nPU	mIoU
Seamless [4]	8	8	8	50.4
HN-OCR-W48 [5]	0.5	16	2	50.8
PDL-X71 [1]	1	64	32	55.4
SNpyr-RN18	0.6	24	1	46.3
SNpyr-RN34	0.6	22	1	49.4
SNpyr-RN152	0.6	8	1	50.1
SNpyr-DN121	0.6	22	1	49.5
SNpyr-DN161	0.6	17	1	52.2

Table 2: Validation of semantic segmentation architectures. All methods use single-scale evaluation, and train on Tesla V100 32GB GPUs except PDL which trains on TPUs. Pyramidal Swiftnets achieve competitive performance with respect to the state of the art, while requiring much less computational resources.

The table shows that pyramidal SwiftNets offer competitive generalization performance under modest computational requirements. This makes them a good choice for large-scale experiments such as RVC 2020. Furthermore, larger models require significantly more time to complete the training, which may make a difference in large-scale multi-domain settings.

We observe that increasing the model capacity results in diminishing returns (cf. SNPyr-RN18 vs SNPyr-RN34 vs SNPyr-RN152). SNpyr-DN161 outperforms SNPyr-RN152 in spite a slightly weaker backbone. We speculate that this may be due to larger batches.

2.2. Validation of multi-scale input

Table 3 compares a pyramidal model (SNPyr-DN161s3) to its single-scale counterpart (SN-DN161s3). Both models are based on DenseNet-161 with 64× subsampled representation at the far end which we achieve by splitting the 3rd dense block [2]. The single scale model (SN-DN161s3) uses an SPP module at the end of the downsampling path at 64× subsampled resolution.

2.3. Validation of upsampling width

Table 4 shows the influence of the upsampling width on the segmentation performance. Wider upsampling improves the accuracy, although not significantly.

Model	MPx	BS	nPU	mIoU
SNpyr-DN161s3	0.6	18	1	**50.6**
SN-DN161s3	0.6	25	1	48.1

Table 3: Validation of pyramidal fusion. The multi-resolution model (SNpyr-DN161s3) outperforms its single-scale counterpart (SN-DN161s3).

Model	MPx	BS	nPU	mIoU
SNpyr-DN161s3	0.6	18	1	50.6
SNPyr-DN161s3-fat	0.6	18	1	**50.9**

Table 4: Validation of the width of the upsampling path. Wider upsampling path achieves a slightly better accuracy.

2.4. Validation of block splitting

Table 5 examines the influence of dense-block splitting to the generalization performance. We observe that splitting the 3rd block deteriorates the segmentation performance when pyramidal organization is used. We speculate that this is due to too much subsampling within the feature extractor.

Model	MPx	BS	nPU	mIoU
SNpyr-DN161	0.6	17	1	**52.2**
SNpyr-DN161s3	0.6	18	1	50.6

Table 5: Validation of the extent of subsampling within a multi-scale feature extractor. Splitting the third block reduces segmentation accuracy.

3. Qualitative universal performance

Figure 1 extends Figure 2 from the main paper by including qualitative universal performance of our SNp-DN161 model. The color map for the universal label space combines the color maps from Vistas and ADE20k. We observe that the model is able to recognize refinements of concepts from particular datasets. For instance the model locates road markings and sidewalk curbs in images from Cityscapes, VIPER, KITTI and ADE20k. It also succeeds to discriminate bushes and trees in KITTI and WildDash 2 although this distinction exists only in the ADE20k taxonomy.

Figure 2 shows predictions of universal classes which correspond to the class road in Cityscapes test. Formally, the set of all such classes corresponds to $m_{\mathbb{S}_{\mathrm{CS}}}(\mathrm{CS}-\mathrm{road})$.

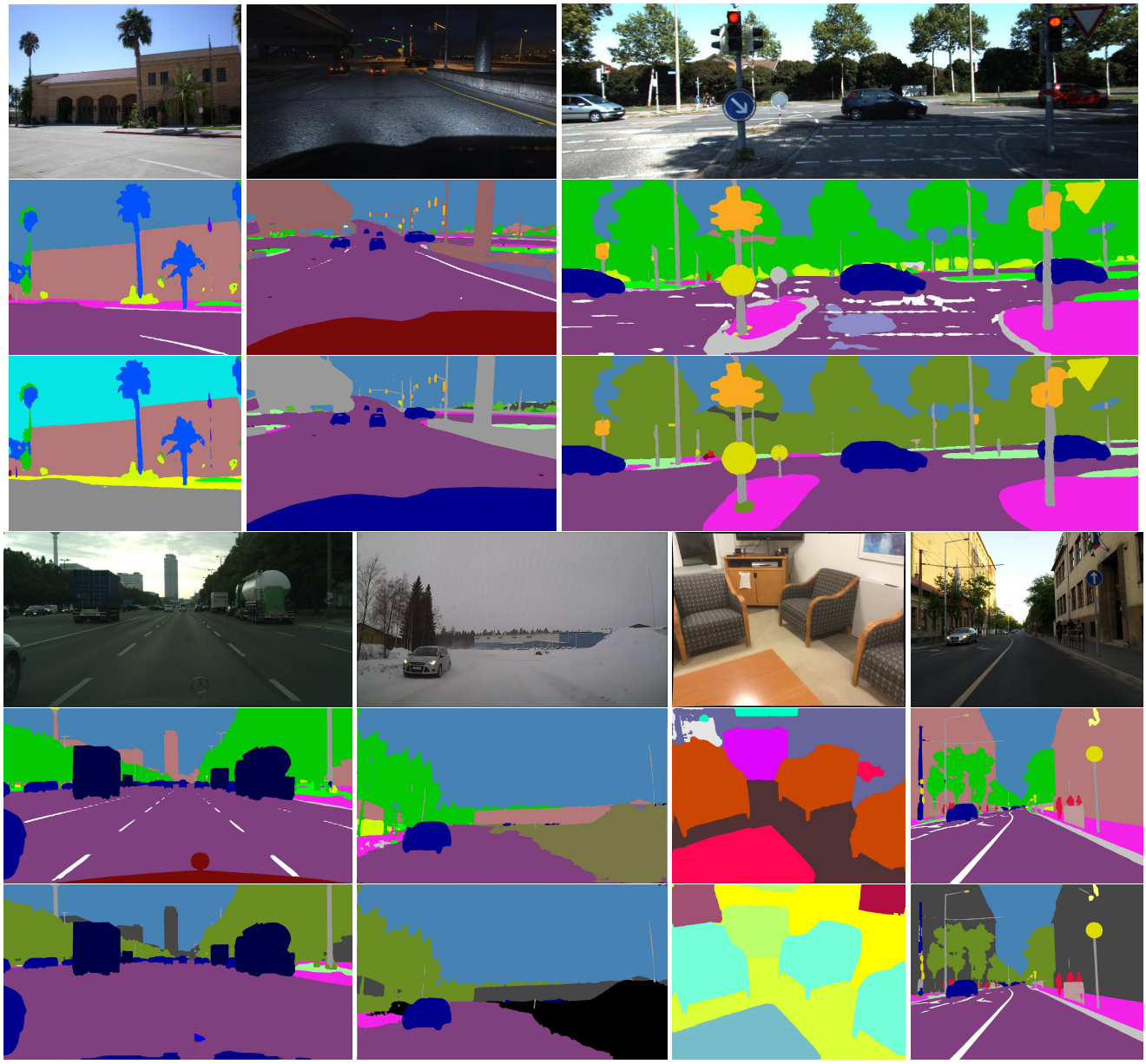


Figure 1: Qualitative performance of our universal SNp-DN161 model on test images from the seven RVC datasets. Rows 1 and 4 show input images, rows 2 and 5 show predictions in the universal label space, while rows 3 and 6 show dataset-specific predictions. Images belong to (top to bottom, left to right) ADE20k, Viper, KITTI, Cityscapes, WildDash, ScanNet and Vistas. Universal predictions find classes which are not labeled in the corresponding dataset, eg. crosswalk, curb and road-marking in the image from KITTI. All road driving datasets represent vegetation with a single class, but the universal model is able to classify those pixels more precisely as trees, palms and plants.

These universal classes are: road, bike lane, crosswalk, zebra, road marking, pothole, manhole and service lane. No instance of pothole was found in the entire Cityscapes test. The best performing classes are road and road marking. We observe recognition of zebras only at close range, while otherwise they often get classified as road marking. The remaining classes are usually detected as small clusters in the correct region. For example, manholes are often only partially segmented.

Figure 3 shows the performance of our model on negative images from WildDash. These images were taken in non-road driving contexts (rows 1-4) or from an unusual

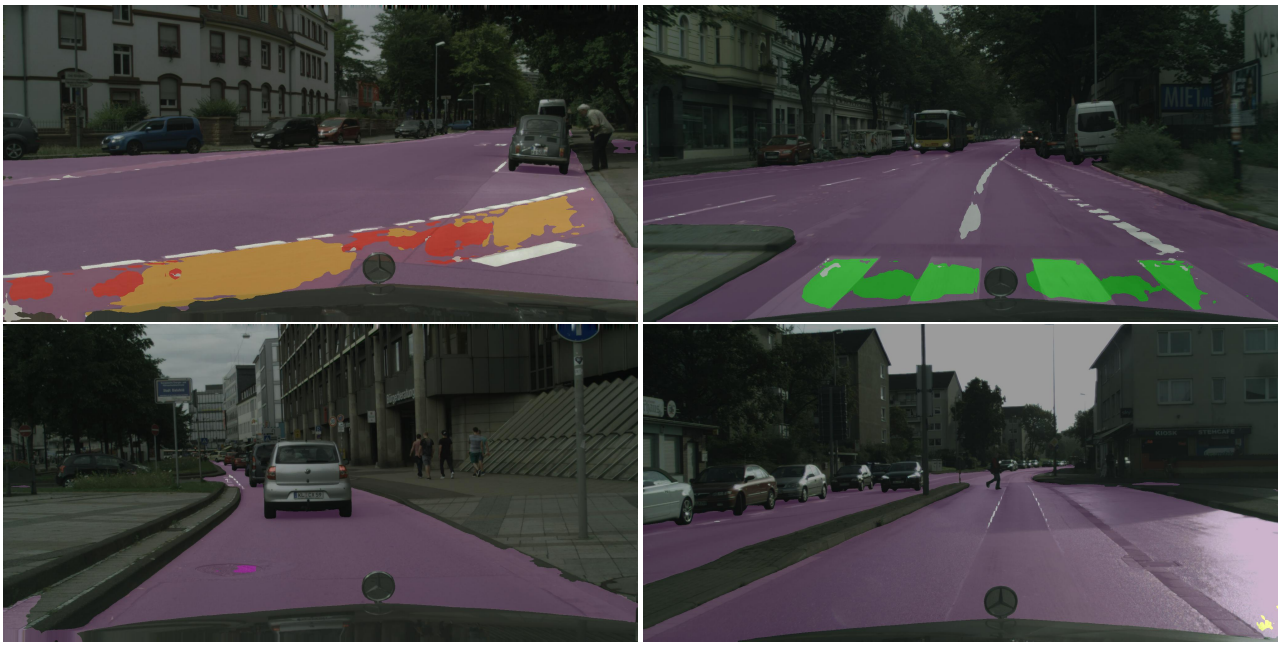


Figure 2: Qualitative performance of our universal SNp-DN161 model on Cityscapes test. We overlay predictions over the input images. We show predictions of following universal classes: road (purple, all images), bike lane (yellow, top-left), crosswalk (red, top-left), zebra (green, top-right), road marking (white, all images), manhole (magenta, bottom-left) and service lane (lemon yellow, bottom-right).

perspective (row 5). These images may contain classes found in traffic scenes such as people (row 2). Note that the benchmark accepts either the best-case ground truth or the void class (denoted with black).

Figure 4 shows an interesting failure of our model to distinguish between different types of animals. Our universal model classifies the horse from an ADE20k image as class bird. This occurs since ADE20k taxonomy contains only the class animal which we map to universal classes bird and ground-animal. The training signal for distinguishing between birds and ground animals should have come from Vistas, but this signal was very weak since these two Vistas classes are extremely rare.

4. Mapping visualizations

Figures 5 and 6 visualize training and evaluation mappings used in our City-Vistas experiments. We show these mappings for the two baselines and our universal taxonomy. Please refer to Section 3 of the main paper for a detailed description of the procedure for recovering these mappings.

References

[1] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. 2

[2] Ivan Krešo, Josip Krapac, and Siniša Šegvić. Efficient ladder-style densenets for semantic segmentation of large images. *IEEE Transactions on Intelligent Transportation Systems*, 2020. 2

[3] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020. 1

[4] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, pages 8277–8286, 2019. 2

[5] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 2

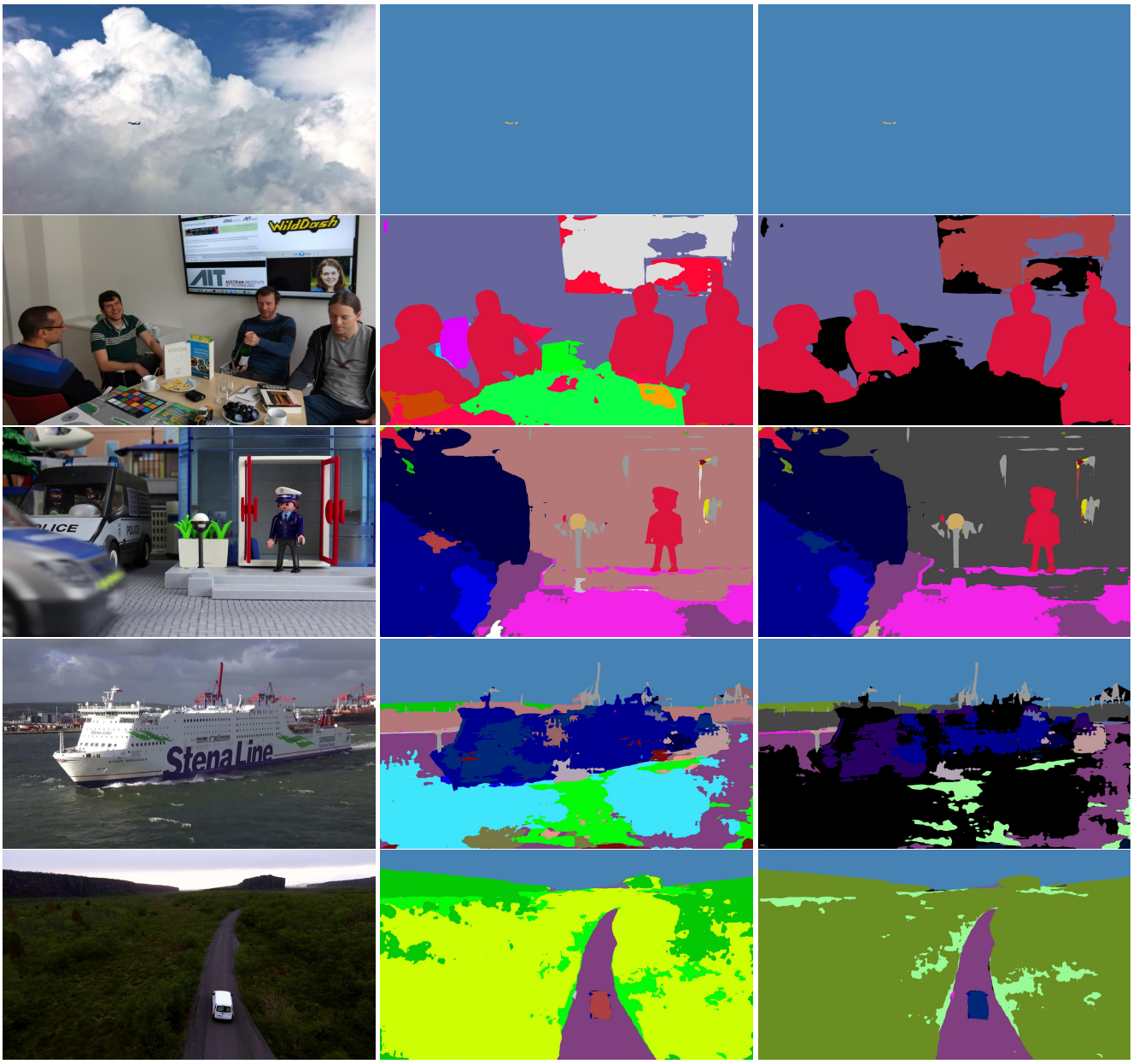


Figure 3: Performance of our universal model on four WildDash 2 negative test images. The columns show the input image, universal segmentation, and the segmentation in the WildDash label space where class void is shown in black. Our model successfully recognizes some non-traffic classes, e.g. table, chair, book, and cabinet (row 2), or boat and water (row 5). The model is robust to perspective changes (row 5) and exhibits fair performance in presence of large domain-shift (row 3).

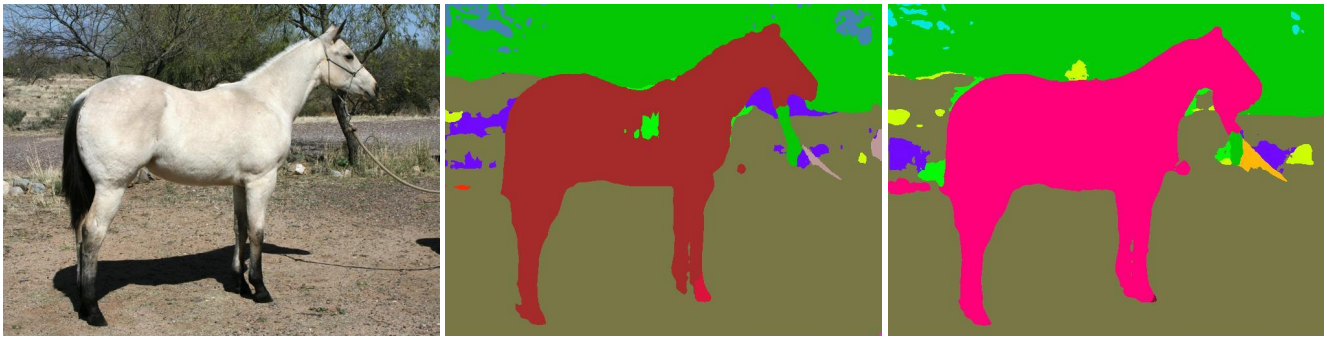
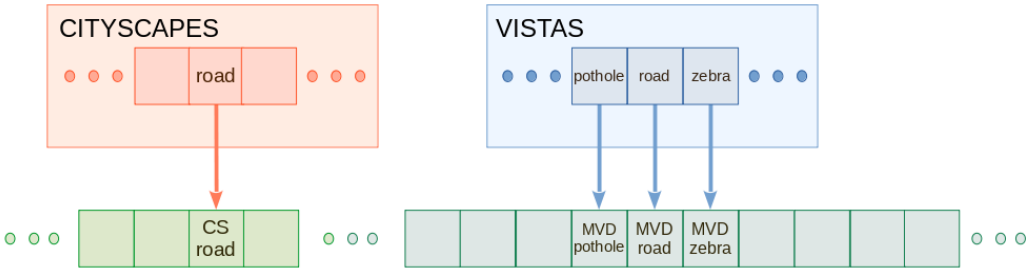
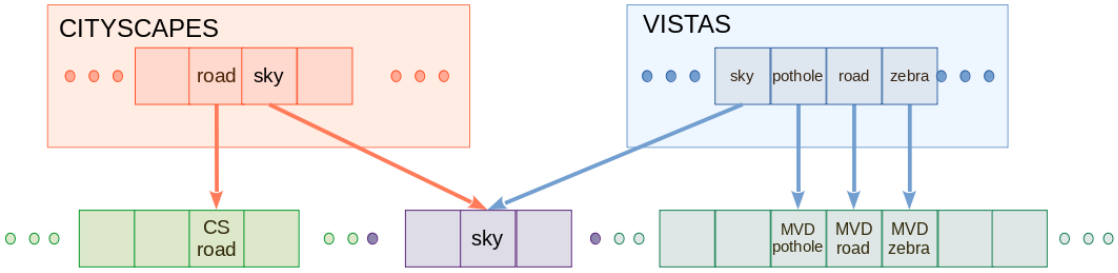


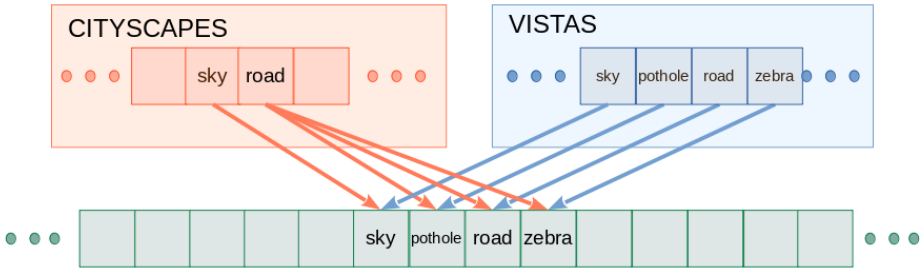
Figure 4: A failure case on ADE20k test. The columns show the input image, universal segmentation, and ADE20k segmentation. The model recognizes most of the horse as class bird. This occurs since birds and ground animals are annotated only in Vistas as extremely rare classes. Note that two patches are incorrectly classified as grass (green) and pedestrian (red). These patches finaly get correctly classified into ADE20k-animal, since the sum of probabilities of classes bird and ground animal prevails after evaluation mapping.



(a)

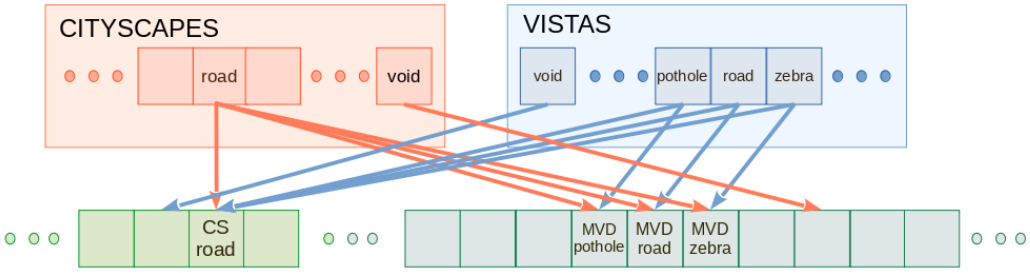


(b)

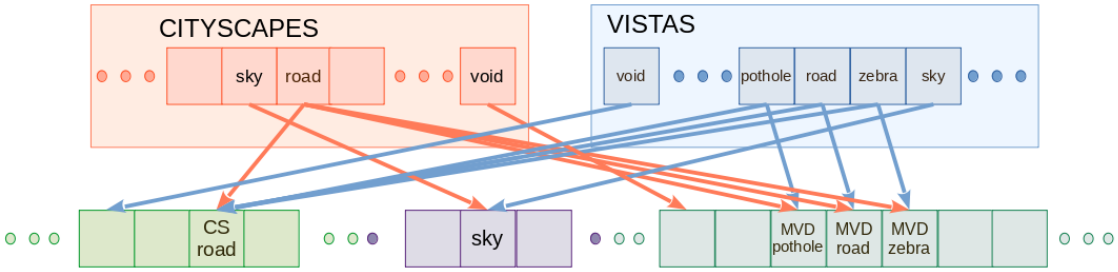


(c)

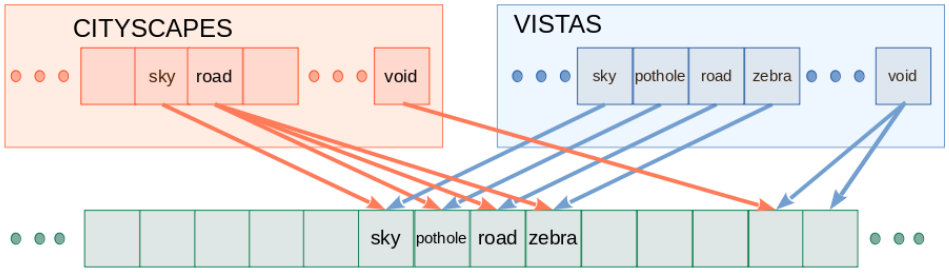
Figure 5: Visualizations of training mappings in City-Vistas experiments. Naive concatenation (a) maps each dataset-specific class to the corresponding training logit. Partial merge (b) maps dataset-specific classes to the common logit only if they match exactly. If this is not the case, the classes remain separate. Our universal taxonomy (c) maps each dataset-specific class to one or more universal classes.



(a)



(b)



(c)

Figure 6: Visualizations of evaluation mappings in City-Vistas experiments for naive concatenation (a), partial merge (b) and universal taxonomy (c). In each of these cases, dataset-specific classes are mapped to training classes with which they overlap. We extend dataset-specific taxonomies with a void class that maps to all training classes which do not overlap with any of dataset-specific classes.