MSeg: A Composite Dataset for Multi-domain Semantic Segmentation

John Lambert* [1,3], Zhuang Liu*[1,2], Ozan Sener[1], James Hays[3,4], and Vladlen Koltun[1]

[1]Intel Labs, [2]University of California, Berkeley, [3]Georgia Institute of Technology, [4]Argo AI

Figure 1: MSeg unifies multiple semantic segmentation datasets by reconciling their taxonomies and resolving incompatible annotations. This enables training models that perform consistently across domains and generalize better. Input images in this figure were taken (top to bottom) from the ScanNet [8], WildDash [44], and Pascal VOC [10] datasets, none of which were seen during training.

Abstract

We present MSeg, a composite dataset that unifies semantic segmentation datasets from different domains. A naive merge of the constituent datasets yields poor performance due to inconsistent taxonomies and annotation practices. We reconcile the taxonomies and bring the pixel-level annotations into alignment by relabeling more than 220,000 object masks in more than 80,000 images. The resulting composite dataset enables training a single semantic segmentation model that functions effectively across domains and generalizes to datasets that were not seen during training. We adopt zero-shot cross-dataset transfer as a benchmark to systematically evaluate a model's robustness and show that MSeg training yields substantially more robust models in comparison to training on individual datasets or naive mixing of datasets without the presented contributions. A model trained on MSeg ranks first on the WildDash leaderboard for robust semantic segmentation, with no exposure to WildDash data during training.

*Equal contribution

1. Introduction

When Papert first proposed computer vision as a summer project in 1966 [26], he described the primary objective as "...a system of programs which will divide a vidisector picture into regions such as likely objects, likely background areas and chaos." Five decades later, computer vision is a thriving engineering field, and the task described by Papert is known as semantic segmentation [5, 15, 20, 33, 42, 45].

Have we delivered on Papert's objective? A cursory examination of the literature would suggest that we have. Hundreds of papers are published every year that report ever-higher accuracy on semantic segmentation benchmarks such as Cityscapes [7], Mapillary [25], COCO [19], ADE20K [46], and others. Yet a simple exercise can show that the mission has not been accomplished. Take a camera and begin recording as you traverse a sequence of environments: for example, going about your house to pack some supplies, getting into the car, driving through your city to a forest on the outskirts, and going on a hike. Now perform semantic segmentation on the recorded video. Is there a model that will successfully perform this task?

A computer vision professional will likely resort to mul-

tiple models, each trained on a different dataset. Perhaps a model trained on the NYU dataset for the indoor portion [34], a model trained on Mapillary for the driving portion, and a model trained on ADE20K for the hike. Yet this is not a satisfactory state of affairs. It burdens practitioners with developing multiple models and implementing a controller that decides which model should be used at any given time. It also indicates that we haven't yet arrived at a satisfactory vision system: after all, an animal can traverse the same environments with a single visual apparatus that continues to perform its perceptual duties throughout.

A natural solution is to train a model on multiple datasets, hoping that the result will perform as well as the best dedicated model in any given environment. As has previously been observed, and confirmed in our experiments, the results are far from satisfactory. A key underlying issue is that different datasets have different taxonomies: that is, they have different definitions of what constitutes a 'category' or 'class' of visual entities. Taxonomic clashes and inconsistent annotation practices across datasets from different domains (e.g., indoor and outdoor, urban and natural, domain-specific and domain-agnostic) substantially reduce the accuracy of models trained on multiple datasets.

In this paper, we take steps towards addressing these issues. We present MSeg, a composite dataset that unifies semantic segmentation datasets from different domains: COCO [19], ADE20K [46], Mapillary [25], IDD [40], BDD [43], Cityscapes [7], and SUN RGB-D [36]. A naive merge of the taxonomies of the seven datasets would yield more than 300 classes, with substantial internal inconsistency in definitions and annotation standards. Instead, we reconcile the taxonomies, merging and splitting classes to arrive at a unified taxonomy with 194 categories. To bring the pixel-level annotations in conformance with the unified taxonomy, we conduct a large-scale annotation effort via the Mechanical Turk platform and produce compatible annotations across datasets by relabeling object masks.

The resulting composite dataset enables training unified semantic segmentation models that come a step closer to delivering on Papert's vision. MSeg training yields models that exhibit much better generalization to datasets that were not seen during training. We adopt *zero-shot cross-dataset transfer* as a proxy for a model's expected performance in the "real world" [27]. In this mode, MSeg training is substantially more robust than training on individual datasets, or training on multiple datasets without the reported taxonomic reconciliation. In particular, our MSeg-trained model sets a new state of the art of the WildDash benchmark for robust semantic segmentation [44]. Our model ranks first on the WildDash leaderboard, without seeing any WildDash data during training.

2. Related Work

Cross-domain semantic segmentation. Mixing segmentation datasets has primarily been done within a single domain and application, such as driving. Ros et al. [30] aggregated six driving datasets. Bevandic et al. [1] mix Mapillary Vistas, Cityscapes, the WildDash validation set, and ImageNet-1K-BB (a subset of ImageNet [9] for which bounding box annotations are available) for joint segmentation and outlier detection on WildDash [44]. On a smaller scale, [16, 22] mix Mapillary, Cityscapes, and the German Traffic Sign Detection Benchmark. In contrast to these works, we focus on semantic segmentation across multiple domains and resolve inconsistencies between datasets at a deeper level, including relabeling incompatible annotations.

Varma et al. [40] evaluate the transfer performance of semantic segmentation datasets for driving. They only use 16 common classes, without any dataset mixing. They observe that cross-dataset transfer is significantly inferior to "self-training" (i.e., training on the target dataset). We have observed the same outcomes when models are trained on individual datasets, or when datasets are mixed naively.

Liang et al. [18] train a model by mixing Cityscapes, ADE20K, COCO Stuff, and Mapillary, but do not evaluate cross-dataset generalization. Kalluri et al. [14] mix pairs of datasets (Cityscapes + CamVid, Cityscapes + IDD, Cityscapes + SUN RGB-D) for semi-supervised learning.

An underlying issue that impedes progress on unified semantic segmentation is the incompatibility of dataset taxonomies. In contrast to the aforementioned attempts, we directly address this issue by deriving a consistent taxonomy that bridges datasets from multiple domains.

Domain adaptation and generalization. Training datasets are biased and deployment in the real world presents the trained models with data that is unlike what had been seen during training [38]. This is known as *covariate shift* [32] or *selection bias* [13], and can be tackled in the *adaptation* or the *generalization* setting. In *adaptation*, samples from the test distribution (deployment environment) are available during training, albeit without labels. In *generalization*, we expect models to generalize to previously unseen environments after being trained on data from multiple domains.

We operate in the generalization mode and aim to train robust models that perform well in new environments, with no data from the target domain available during training. Many domain generalization approaches are based on the assumption that learning features that are invariant to the training domain will facilitate generalization to new domains [21, 23]. Volpi et al. [41] use distributionally robust optimization by considering domain difference as noise in the data distribution space. Bilen and Vedaldi [2] propose to learn a unified representation and eliminate domain-specific scaling factors using instance normalization. Mancini et al. [21] modify batch normalization statistics to make fea-

tures and activations domain-invariant.

The aforementioned domain generalization methods assume that the same classifier can be applied in all environments. This relies on compatible definitions of visual categories. Our work is complementary and can facilitate future research on domain generalization by providing a compatible taxonomy and consistent annotations across semantic segmentation datasets from different domains.

Visual learning over diverse domains. The Visual Domain Decathlon [28] introduced a benchmark over ten image classification datasets, but allows training on all of them. More importantly, its purpose is not training a single classifier. Instead, they hope domains will assist each other by transferring inductive biases in a multi-task setting. Triantafillou et al. [39] proposed a meta-dataset for benchmarking few-shot classification algorithms.

For the problem of monocular depth estimation, Ranftl et al. [27] use multiple datasets and mix them via a multi-task learning framework. We are inspired by this work and aim to facilitate progress on dataset mixing and cross-dataset generalization in semantic segmentation. Unlike the work of Ranftl et al., which dealt with a geometric task (depth estimation), we are confronted with inconsistencies in semantic labeling across datasets, and make contributions towards resolving these.

3. The MSeg Dataset

Table 1 lists the semantic segmentation datasets used in MSeg. This set of datasets is the result of a selection process that considered a much larger number of candidates. The datasets that were not used, and reasons for not including them, are listed in the supplement.

Our guiding principle for selecting a training/test dataset split is that large, modern datasets are most useful for training, whereas older and smaller datasets are good candidates for testing. We test zero-shot cross-dataset performance on the validation subsets of these datasets. Note that data from the test datasets (including their training splits) is never used for training in MSeg. For validation, we use the validation subsets of the training datasets listed in Table 1.

We use the free, academic version of Mapillary Vistas [25]. In this we forego highly detailed classification of traffic signs, traffic lights, and lane markings in favor of broader access to MSeg.

For COCO [19], we use the taxonomy of COCO Panoptic as a starting point, rather than COCO Stuff [4]. The COCO Panoptic taxonomy merges some of the material-based classes of COCO Stuff into common categories that are more compatible with other datasets. (E.g., *floor-marble*, *floor-other*, and *floor-tile* are merged into *floor*.)

Naively combining the component datasets yields roughly 200K images with 316 semantic classes (after merging classes with synonymous names). We found that training on naively combined datasets yields low accuracy and poor generalization. We believe the main cause for this failure is inconsistency in the taxonomies and annotation practices in the different datasets. The following subsections explain these issues and our solution.

Table 1: Component datasets in MSeg.

Dataset name	Origin domain	# Images
Training & Validation		
COCO [19] + COCO STUFF [4]	Everyday objects	123,287
ADE20K [46]	Everyday objects	22,210
MAPILLARY [25]	Driving (Worldwide)	20,000
IDD [40]	Driving (India)	7,974
BDD [43]	Driving (United States)	8,000
CITYSCAPES [7]	Driving (Germany)	3,475
SUN RGBD [36]	Indoor	5,285
Test		
PASCAL VOC [10]	Everyday objects	1,449
PASCAL CONTEXT [24]	Everyday objects	5,105
CAMVID [3]	Driving (U.K.)	101
WILDDASH [44]	Driving (Worldwide)	70
KITTI [11]	Driving (Germany)	400
SCANNET-20 [8]	Indoor	5,436

3.1. Taxonomy

In order to train a cross-domain semantic segmentation model, we need a unified taxonomy. We followed a sequence of decision rules, summarized in Figure 3, to decide on split and merge operations on taxonomies of the component datasets. We condensed the 316 classes obtained by merging the component datasets into a unified taxonomy of 194 classes. The full list is given in Figure 4 and further described and visualized in the supplement. Each of these classes is derived from classes in the component datasets.

We have two primary objectives in designing the MSeg taxonomy. First, as many classes should be preserved as possible. For example, *guardrail* should not be discarded just because COCO, BDD, or IDD do not annotate it. Merging classes can reduce the discriminative ability of the resulting models. Second, the taxonomy should be flat, rather than hierarchical, to maximize compatibility with standard training methods.

An MSeg category can have one of the following relationships to classes in a component dataset: (a) it can be in direct correspondence to a class in a component taxonomy, (b) it can be the result of merging a number of classes from a component taxonomy, (c) it can be the result of splitting a class in a component taxonomy (one-to-many mapping), or (d) it can be the union of classes which are split from different classes in the component taxonomy.

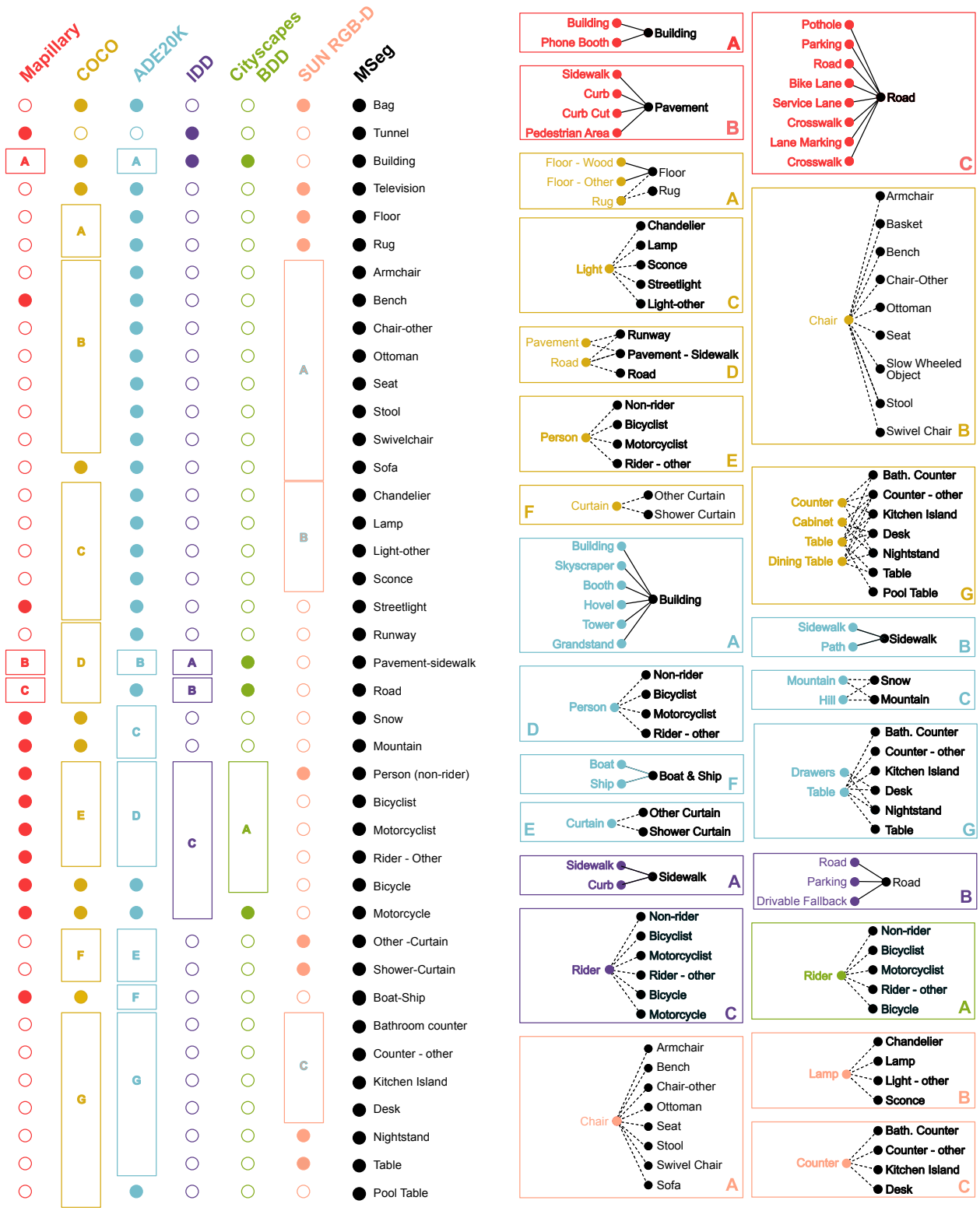

Figure 2: Visualization of a subset of the class mapping from each dataset to our unified taxonomy. This figure shows 40 of the 194 classes; see the supplement for the full list. Each filled circle means that a class with that name exists in the dataset, while an empty circle means that there is no pixel from that class in the dataset. A rectangle indicates that a split and/or merge operation was performed to map to the specified class in MSeg. Rectangles are zoomed-in in the right panel. Merge operations are shown with straight lines and split operations are shown with dashed lines. *(Best seen in color.)*

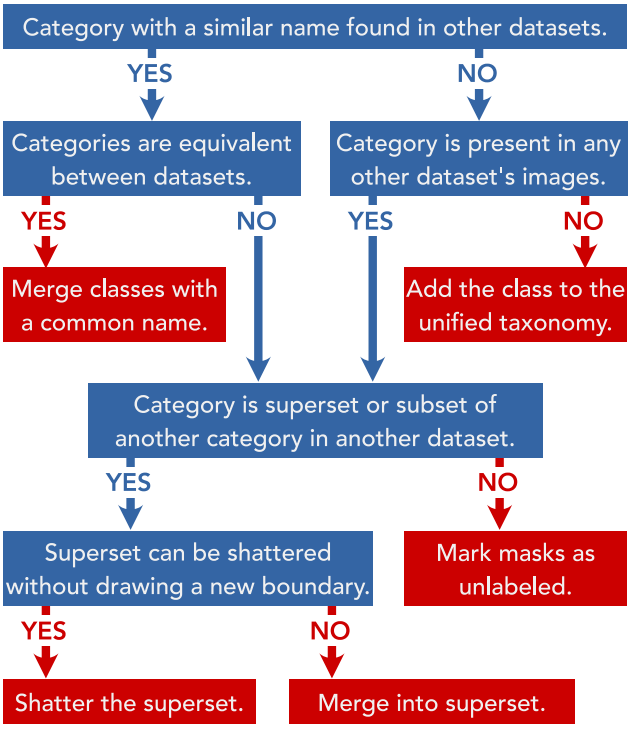

Figure 3: Procedure for determining the set of categories in the MSeg taxonomy. See the supplement for more details.

Figure 2 visualizes these relationships for 40 classes. For example, the class 'person' in COCO and ADE20K corresponds to four classes ('person', 'rider-other', 'bicyclist', and 'motorcyclist') in the Mapillary dataset. Thus the 'person' labels in COCO and ADE20K need to be split into one of the aforementioned four Mapillary categories depending on the context. (See boxes COCO-E and ADE20K-D in Figure 2.) Mapillary is much finer-grained than other driving datasets and classifies *Pothole, Parking, Road, Bike Lane, Service Lane, Crosswalk-Plain, Lane Marking-General, Lane Marking-Crosswalk* separately. These classes are merged into a unified MSeg 'road' class. (See box Mapillary-C in Figure 2.)

Merging and splitting of classes from component datasets have different drawbacks. Merging is easy and can be performed programmatically, with no additional labeling. The disadvantage is that labeling effort that was invested into the original dataset is sacrificed and the resulting taxonomy has coarser granularity. Splitting, on the other hand, is labor-intensive. To split a class from a component dataset, all masks with that class need to be relabeled. This provides finer granularity for the resulting taxonomy, but costs time and labor. The procedure summarized in Figure 3 is our approach to trading off these costs.

3.2. Relabeling Instances of Split Classes

We utilize Amazon Mechanical Turk (AMT) to relabel masks of classes that need to be split. We re-annotate only the datasets used for learning, leaving the evaluation datasets intact. Instead of recomputing boundaries, we formulate the problem as multi-way classification and ask annotators to classify each mask into finer-grained categories from the MSeg taxonomy. We include an example labeling screen, workflow and labeling validation process in the supplement. In total, we split 31 classes and relabel 221,323 masks. We visualize some of the split operations in Figure 2 and provide additional details in the supplement.

AMT workers sometimes submit inaccurate, random, or even adversarial decisions [35]. To ensure annotation quality, we embed 'sentinel' tasks within each batch of work [6, 12, 29], constituting at least 10% of each batch. These sentinels are tasks for which the ground truth is unambiguous and is manually annotated by us. We use the sentinels to automatically evaluate the reliability of each annotator so that we can direct work towards more reliable annotators. Five workers annotate each batch, and the work is resubmitted until all submitted batches meet a 100% sentinel accuracy. Afterwards, the category is determined by majority vote; categories that do not meet these criteria are manually labeled in-house by expert annotator (one of the authors).

4. Experimental Results

Implementation details. We use the HRNet-W48 [37] architecture as our model. We use SGD with momentum and polynomial learning rate decay, starting with a learning rate of 0.01. When forming a minibatch of size m from multiple datasets, we evenly split the minibatch by the number of training datasets n, meaning each dataset will contribute m/n examples to each minibatch. Accordingly, there is no notion of "epoch" for the unified dataset during our training, but rather only total samples seen from each dataset. For example, in a single effectual "COCO epoch", Mapillary will complete more than 6 effectual epochs, as its dataset is less than $\frac{1}{6}$th the size of COCO. We train until one million crops from each dataset's images have been seen.

Image resolution is inconsistent across component datasets. For example, Mapillary contains many images of resolution $\sim 2000 \times 4000$, while most ADE20K images have resolution $\sim 300 \times 400$. Before training, we use $2\times$ or $3\times$ super-resolution [17] to first upsample the training datasets with lower resolution to a higher one (at least 1000p). At training time, we resize images from different datasets to a consistent resolution. Specifically, in our experiments, we resize all images such that their shorter side is 1080 pixels (while preserving aspect ratios) and use a crop size of 713×713px. At test time, we resize the image to one of three different resolutions ($360/720/1080$ as the images' shorter side), perform inference, and then interpolate the prediction maps back to the original resolutions for evaluation. The resolution level ($360/720/1080$) is set per dataset. More details are provided in the supplement.

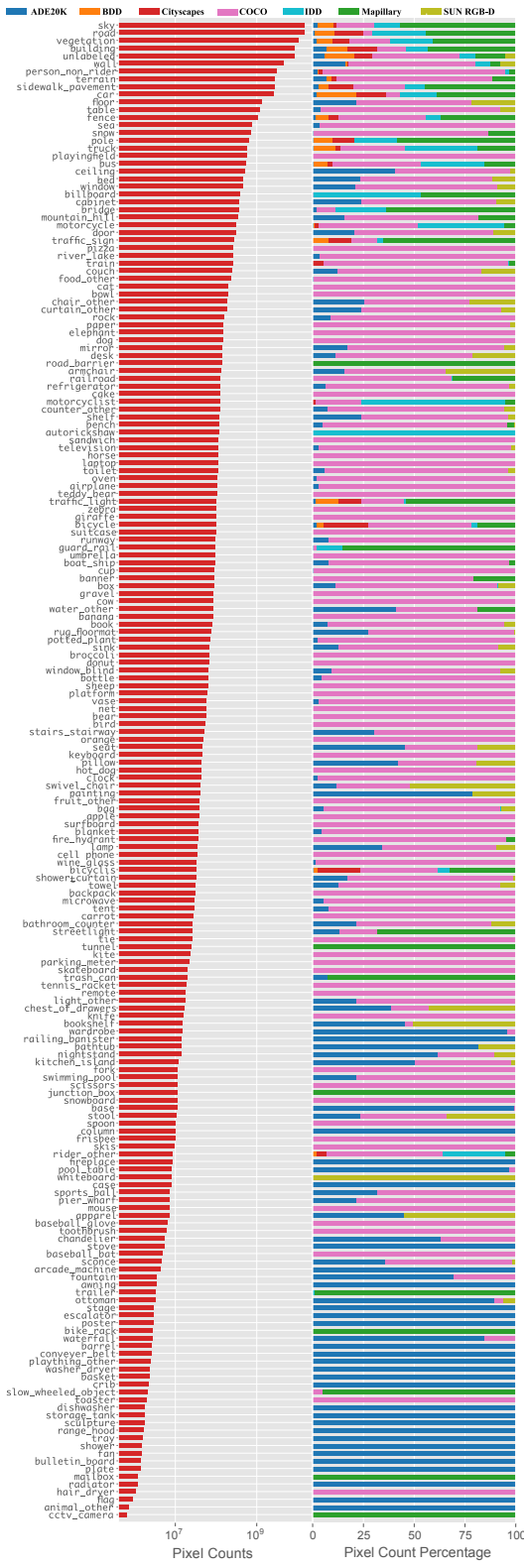

Figure 4: Semantic classes in MSeg. Left: pixel counts of MSeg classes, in log scale. Right: percentage of pixels from each component dataset that contribute to each class. Any single dataset is insufficient for describing the visual world.

Using the MSeg taxonomy on a held-out dataset. At inference time, at each pixel we obtain a vector of probabilities over the unified taxonomy's m_u categories. These unified taxonomy probabilities must be allocated to test dataset taxonomy buckets. For example, we have three separate probabilities in our unified taxonomy for 'motorcyclist', 'bicyclist', and 'rider-other'. We sum these three together to compute a Cityscapes 'rider' probability. We implement this remapping from m_u classes to m_t classes for the evaluation dataset as a linear mapping P from $\mathbb{R}^{m_u}$ to $\mathbb{R}^{m_t}$. The matrix weights P_{ij} are binary 0/1 values and are fixed before training or evaluation; the weights are determined manually by inspecting label maps of the test datasets. P_{ij} is set to 1 if unified taxonomy class j contributes to evaluation dataset class i, otherwise $P_{ij} = 0$.

Zero-shot transfer performance. We use the MSeg training set to train a unified semantic segmentation model. Table 2 lists the results of zero-shot transfer of the model to MSeg test datasets. Note that none of these datasets were seen by the model during training. For comparison, we list the performance of corresponding models that were trained on the individual training datasets that were used to make up MSeg. For reference, we also list the performance of 'oracle' models that were trained on the training splits of the test datasets. Note that WildDash does not have a training set, thus no 'oracle' performance is provided for it.

The results in Table 2 indicate that good performance on a particular test dataset can sometimes be obtained by training on a specific training dataset that has compatible priors. For example, training on COCO yields good performance on VOC, and training on Mapillary yields good performance on KITTI. But no individual training dataset yields good performance across test datasets. In contrast, the model trained on MSeg performs consistently across all datasets. This is evident in the aggregate performance, summarized by the harmonic mean across datasets. The harmonic mean mIoU achieved by the MSeg-trained model is 28% higher than the accuracy of the best individually-trained baseline (COCO).

Performance on training datasets. Table 3 lists the accuracy of trained models on the MSeg training datasets. We test on the validation sets and compute IoU on a subset of classes that are jointly present in the dataset and MSeg's taxonomy. Except for Cityscapes and BDD100K, results on validation sets of all training datasets are not directly comparable to the literature since the MSeg taxonomy involves merging multiple classes. As expected, individually-trained models generally demonstrate good accuracy when tested on the same dataset: a model trained on COCO performs well on COCO, etc. The aggregate performance of the MSeg model is summarized by the harmonic mean across datasets. It is 68% higher than the best individually-trained baseline (COCO).

WildDash benchmark. The WildDash benchmark [44] specifically evaluates the robustness of semantic segmentation models. Images mainly contain road scenes with unusual and hazardous conditions (e.g., poor weather, noise, distortion). The benchmark is intended for testing the robustness of models trained on other datasets, and does not provide a training set of its own. A small set of 70 annotated images is provided for validation. The primary mode of evaluation is a leaderboard, with a testing server and a test set with hidden annotations. The main evaluation measure is Meta Average mIoU, which combines performance metrics associated with different hazards and per-frame IoU.

We submitted result from a model trained on MSeg to the WildDash test server, with multi-scale inference. Note that WildDash in not among the MSeg training sets and the submitted model has never seen WildDash images during training. The results are reported in Table 4. Our model is ranked 1st on the leaderboard. Remarkably, our model outperforms methods that were trained on multiple datasets and utilized the WildDash validation set during training. In comparison to the best prior model that (like ours) did not leverage WildDash data during training, our model improves accuracy by 9.3 percentage points: a 24% relative improvement.

Algorithms for learning from multiple domains. We evaluate the effectiveness of algorithmic approaches for multi-domain learning, specifically domain generalization and multi-task learning. We use a state-of-the-art multi-task learning algorithm [31] and a Domain Generalization (DG) algorithm [23]. The multi-task learning algorithm, MGDA [31], finds a Pareto optimal solution that trades off the losses over the different datasets. The DG baseline, Classification and Contrastive Semantic Alignment (CCSA) [23], enforces representation invariance across datasets.

We compare MGDA and CCSA with our simple strategy of evenly mixing data in Table 5. For this experiment, we used only COCO, Mapillary, and ADE20K, at a reduced resolution (roughly QVGA, shorter image side is 240 px). (We provide the high-resolution result on all 7 training datasets in the supplement.) We find that on the majority of test datasets, multi-task learning slightly hurts the zero-shot transfer performance compared with plain mixing of data from different datasets in a batch. The DG algorithm appears to hurt performance significantly. Additional details are provided in the supplement.

Qualitative results. Figure 5 provides qualitative results on images from different test datasets. Unlike the baselines, the MSeg model is successful in *all* domains. On ScanNet, our model provides more accurate predictions for chairs than even the provided ground truth. In comparison, ADE20K models are blind to tables and Mapillary-trained models completely fail in ScanNet's indoor regime. On CamVid, the Mapillary- and COCO-trained models incorrectly predict sidewalk on the road surface; ADE20K- and COCO-trained models have no notion of rider and mistake bicyclists for pedestrians. On Pascal VOC, our model is the only one to correctly identify a person standing on an airplane's mobile staircase; an ADE20K-trained model erroneously predicts a boat, and a Mapillary model sees a car. On another Pascal image, ADE20K has no horse class, and the corresponding model cannot identify it.

Ablation study. Table 6 reports a controlled evaluation

Table 2: Semantic segmentation accuracy (mIoU) on MSeg test datasets. (Zero-shot cross-dataset generalization.) *Top:* performance of models trained on individual training datasets. *Middle:* the same model trained on MSeg (our result). *Bottom:* for reference, performance of 'oracle' models trained on the test datasets. Numbers within 1% of the best are in bold. The rightmost column is a summary measure: harmonic mean across datasets.

Train/Test	VOC	Context	CamVid	WildDash	KITTI	ScanNet	*h. mean*
COCO	**73.7**	**43.1**	56.6	38.9	48.2	33.9	46.0
ADE20K	34.6	24.0	53.5	37.0	44.3	43.8	37.1
Mapillary	22.0	13.5	**82.5**	55.2	**68.5**	2.1	9.2
IDD	14.5	6.3	70.5	40.6	50.7	1.6	6.5
BDD	13.5	6.9	71.0	52.1	55.0	1.4	6.1
Cityscapes	12.1	6.5	65.3	30.1	58.1	1.7	6.7
SUN RGBD	10.2	4.3	0.1	1.4	0.7	42.2	0.3
MSeg	70.8	**42.9**	**83.1**	**63.1**	63.7	**48.4**	**59.0**
Oracle	77.0	46.0	79.1	–	57.5	62.2	–

Table 3: Semantic segmentation accuracy (mIoU) on MSeg training datasets. (Evaluated on validation sets.) *Top:* performance of models trained on individual datasets. *Bottom:* the same model trained on MSeg (our result). Numbers within 1% of the best are in bold. The rightmost column is harmonic mean across datasets.

Train/Test	COCO	ADE20K	Mapillary	IDD	BDD	Cityscapes	SUN	*h. mean*
COCO	**52.6**	19.6	26.7	31.0	44.1	46.2	29.4	32.1
ADE20K	14.5	**45.3**	24.3	27.0	41.5	44.3	35.3	28.7
Mapillary	6.7	6.2	**53.2**	48.2	60.2	69.7	0.2	1.4
IDD	3.1	3.1	24.3	**64.8**	43.7	50.2	0.6	2.8
BDD	3.7	4.1	24.0	33.9	**63.2**	60.9	0.2	1.5
Cityscapes	3.1	3.1	22.4	31.3	45.0	**77.6**	0.2	1.2
SUN RGBD	3.3	7.1	1.1	1.0	2.2	2.6	43.9	2.2
MSeg	48.6	42.8	51.9	61.8	**63.5**	76.3	**46.1**	**53.9**

Table 4: Results from the WildDash leaderboard at the time of submission. Our model, transferred zero-shot, ranks 1st and outperforms models that utilized WildDash data during training.

	Meta AVG mIoU	Seen WildDash data?
MSeg-1080 (Ours)	**48.3**	✗
LDN BIN-768 [1]	46.9	✓
LDN OE [1]	42.7	✓
DN169-CAT-DUAL	41.0	✓
AHiSS [22]	39.0	✗

Table 5: Comparison of a domain generalization algorithm (CCSA) and a multi-task learning algorithm (MGDA) with a plain mixing strategy.

	VOC	WildDash	CamVid	ScanNet	*h. mean*
CCSA [23]	48.9	36.0	52.4	27.0	39.7
MGDA [31]	**69.4**	39.9	57.5	33.5	46.1
Plain mix	69.2	**43.1**	**63.9**	**34.6**	**48.7**

Figure 5: Qualitative results on images from MSeg test datasets. Zero-shot transfer. From top to bottom: ScanNet-20 (top two rows), WildDash, CamVid, and Pascal VOC (bottom two rows).

of two of our contributions: the unified taxonomy (Section 3.1) and the compatible relabeling (Section 3.2). The 'Naive merge' baseline is a model trained on a composite dataset that uses a naively merged taxonomy in which the classes are a union of all training classes, and each test class is only mapped to an universal class if they share the same name. The 'MSeg (w/o relabeling)' baseline uses the unified MSeg taxonomy, but does not use the manually-relabeled data for split classes (Section 3.2). The model trained on the presented composite dataset ('MSeg') achieves better performance than the baselines.

Table 6: Controlled evaluation of unified taxonomy and mask relabeling. Zero-shot transfer to MSeg test datasets. Both contributions make a positive impact on generalization accuracy.

Train/Test	VOC	Context	CamVid	WildDash	KITTI	ScanNet	*h. mean*
Naive merge	51.9	23.8	56.2	59.7	62.6	43.4	44.5
MSeg w/o relabeling	**70.9**	**42.9**	**83.5**	**64.5**	62.6	44.2	58.0
MSeg	**70.8**	**42.9**	**83.1**	63.1	**63.7**	**48.4**	**59.0**

5. Conclusion

We presented a composite dataset for multi-domain semantic segmentation. To construct the composite dataset, we reconciled the taxonomies of seven semantic segmentation datasets. In cases where categories needed to be split, we performed large-scale mask relabeling via the Mechanical Turk platform. We showed that the resulting composite dataset enables training a unified semantic segmentation model that delivers consistently high performance across domains. The trained model generalizes to previously unseen datasets and is currently ranked first on the WildDash leaderboard for robust semantic segmentation, with no exposure to WildDash data during training. We see the presented work as a step towards broader deployment of robust computer vision systems and hope that it will support future work on zero-shot generalization. Code, data, and trained models are available at https://github.com/mseg-dataset.

References

[1] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In *Pattern Recognition*, 2019. 2, 7

[2] Hakan Bilen and Andrea Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv:1701.07275*, 2017. 2

[3] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recogn. Lett.*, 30(2), 2009. 3

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 3

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 1

[6] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 5

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2, 3

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 1, 3

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[10] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 2010. 1, 3

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 3

[12] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 5

[13] James J Heckman. Sample selection bias as a specification error. *Econometrica: Journal of the Econometric Society*, 1979. 2

[14] Tarun Kalluri, Girish Varma, Manmohan Chandraker, and C.V. Jawahar. Universal semi-supervised semantic segmentation. In *ICCV*, 2019. 2

[15] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 2011. 1

[16] Marco Leonardi, Davide Mazzini, and Raimondo Schettini. Training efficient semantic segmentation CNNs on multiple datasets. In *International Conference on Image Analysis and Processing*, 2019. 2

[17] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *CVPR*, 2019. 5

[18] Xiaodan Liang, Hongfei Zhou, and Eric Xing. Dynamic-structured semantic propagation network. In *CVPR*, 2018. 2

[19] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 2, 3

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[21] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Robust place categorization with deep domain generalization. *IEEE Robotics and Automation Letters*, 3(3), 2018. 2

[22] Panagiotis Meletis and Gijs Dubbelman. Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2, 7

[23] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017. 2, 7

[24] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 3

[25] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 1, 2, 3

[26] Seymour A Papert. The summer vision project. 1966. 1

[27] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv:1907.01341*, 2019. 2, 3

[28] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NIPS*. 2017. 3

[29] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, 2017. 5

[30] Germán Ros, Simon Stent, Pablo F. Alcantarilla, and Tomoki Watanabe. Training constrained deconvolutional networks for road scene semantic segmentation. *arXiv:1604.01545*, 2016. 2

[31] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*. 2018. 7

[32] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2), 2000. 2

[33] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1), 2009. 1

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 2

[35] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. In *Empirical Methods in Natural Language Processing*, 2008. 5

[36] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 2, 3

[37] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv:1904.04514*, 2019. 5

[38] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2

[39] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv:1903.03096*, 2019. 3

[40] Girish Varma, Anbumani Subramanian, Anoop M. Namboodiri, Manmohan Chandraker, and C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019. 2, 3

[41] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*. 2018. 2

[42] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 1

[43] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018. 2, 3

[44] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash - creating hazard-aware benchmarks. In *ECCV*, 2018. 1, 2, 3, 7

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[46] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *IJCV*, 127(3), 2019. 1, 2, 3