Learning to Fuse Things and Stuff

Jie Li*, Allan Raventos*, Arjun Bhargava*, Takaaki Tagawa, Adrien Gaidon
Toyota Research Institute (TRI)
firstname.lastname@tri.global

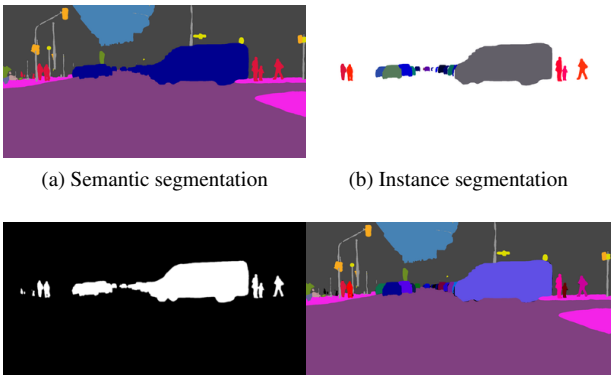
Abstract

We propose an end-to-end learning approach for panoptic segmentation, a novel task unifying instance (things) and semantic (stuff) segmentation. Our model, TASCNet, uses feature maps from a shared backbone network to predict in a single feed-forward pass both things and stuff segmentations. We explicitly constrain these two output distributions through a global things and stuff binary mask to enforce cross-task consistency. Our proposed unified network is competitive with the state of the art on several benchmarks for panoptic segmentation as well as on the individual semantic and instance segmentation tasks.

1. Introduction

Panoptic segmentation is a computer vision task recently proposed by Kirillov et al. [15] that aims to unify the tasks of semantic segmentation (assign a semantic class label to each pixel) and instance segmentation (detect and segment each object instance). This task has drawn attention from the computer vision community as a key next step in dense scene understanding [14, 18, 26], and several publicly available datasets have started to provide labels supporting this task, including Cityscapes [8], Mapillary Vistas [25], ADE20k [42], and COCO [21].

To date, state-of-the-art techniques for semantic and instance segmentation have evolved in different directions that do not seem directly compatible. On one hand, the best semantic segmentation networks [3, 41, 6] focus on dense tensor-to-tensor classification architectures that excel at recognizing *stuff* categories like roads, buildings, or sky [1]. These networks leverage discriminative texture and contextual features, achieving impressive results in a wide variety of scenes. On the other hand, the best performing instance segmentation methods rely on the recent progress in object detection - they first detect 2D bounding boxes of objects and then perform foreground segmentation on regions of interest (RoIs) [27, 12, 23]. This approach uses the key insight that *things* have a well-defined spatial extent and discriminative appearance features.



(a) Semantic segmentation (b) Instance segmentation

(c) Things and Stuff Consistency (d) Panoptic segmentation

Figure 1: We propose an end-to-end architecture for panoptic segmentation. Our model predicts things and stuff with a shared backbone and an internal mask enforcing Things and Stuff Consistency (TASC) that can be used to guide fusion.

The fundamental differences in approaches between handling stuff and things yields a strong natural baseline for panoptic segmentation [15]: using two independent networks for semantic and instance segmentation followed by heuristic post-processing and late fusion of the two outputs.

In contrast, we postulate that addressing the two tasks together will result in increased performance for the joint panoptic task as well as for the separate semantic and instance segmentation tasks.

The basis for this hypothesis is the explicit relation between the tasks at the two ends of the modeling pipeline: i) early on at the feature level (capturing general appearance properties), and ii) at the output space level (mutual exclusion, overlap constraints, and contextual relations).

Therefore, the main challenge we address is how to formulate a unified model and optimization scheme where the sub-task commonalities reinforce the learning, while preventing the aforementioned fundamental differences from leading to training instabilities or worse combined performance, a common problem in multi-task learning [40].

Our main contribution is a deep network and end-to-end learning method for panoptic segmentation that is able to

optimally fuse things and stuff. Most parameters are shared in a ResNet backbone [13] and a 4-stage Feature Pyramid Network (FPN) [20] that is able to learn representations useful for subsequent semantic and instance segmentation heads. In addition, we propose a novel differentiable Things and Stuff Consistency (TASC) to maintain alignment between the output distributions of the two sub-tasks during training. This additional objective encourages separation between the outputs of our semantic and instance segmentation heads to be minimal, while simultaneously enabling mask-guided fusion (cf. Figure 1).

Our unified architecture, TASCNet, maintains or improves the performance of individually trained models and is competitive with Panoptic Quality (PQ) benchmarks on the Mapillary Vistas [25] and Cityscapes [9] datasets. We conduct a detailed ablative analysis, experimentally confirming that our cross-task constraint is key to improving training stability and accuracy. Finally, we show that using a single network has the benefit of simplifying training and inference procedures, while improving efficiency by greatly reducing the number of parameters.

2. Related Work

Tackling dense scene understanding and individual object recognition simultaneously has a long and rich history in computer vision. Tu et al. [36] proposed a hierarchical probabilistic graphical model for scene parsing, disentangling objects, faces, textures, segments, and shapes. This seminal paper inspired a fertile research direction, including contributions on how to explicitly model the relations between things and stuff categories [32, 39, 35, 34, 24]. For instance, Sun et al. [32] use a CRF over image segments to model geometric and semantic relations. Yao et al. [39] incorporate segmentation unary potentials and object reasoning ones (co-occurrence and detection compatibility) in a holistic structured loss. Tighe et al. [34] combined semantic segmentation with per-exemplar sliding window. These approaches rely on handcrafting specific unary and pairwise potentials acting as constraining priors on scene components and their expected relations.

In contrast, deep neural networks can learn powerful shared representations from data, leading to the state of the art in both semantic segmentation [3, 41, 6] and object detection [29, 27, 12, 23]. As the corresponding architectures share fundamental similarities inherited from the seminal AlexNet model [17], several works have naturally leveraged the commonalities to propose multi-task models that can simultaneously address semantic segmentation, object detection, and more [37, 14, 16, 26, 33]. These networks typically follow an encoder-decoder architecture, sharing initial layers followed by separate task-specific branches. In the case of tasks partially competing with each other (e.g., disagreeing on specific image regions), this can result in worse performance (globally and for each task), training instabilities, or outputs not consistent across tasks as noted in [15]. In order to better leverage task affinities and reduce the need for supervision, Zamir et al. [40] build a "taskonomy" by learning general task transfer functions. Other works have proposed simple methods tackling the issue of loss weighting. Kendall et al. [14] propose to use task-dependent uncertainty to weigh the different loss components. Chen et al. [7] propose another weighting based on gradient norms. Alternatively, Sener et al. [31] formulate multi-task learning as a multi-objective optimization problem. These approaches avoid complicated hyper-parameter tuning and reduce training times, but only marginally improve joint performance, under-performing larger individual per-task models.

In contrast to these general multi-task learning approaches, we focus explicitly on the relations between stuff and thing categories, with the goal of improving individual performance and addressing the unified panoptic prediction task. Dai et al. [10] predict things and stuff segmentation with a shared feature extractor and convolutional feature masking of region proposals designed initially for object detection. Those are sampled and combined to provide sufficient coverage of the stuff regions, but the relation between things and stuff is not explicitly leveraged. Chen et al. [5] leverage semantic segmentation logits to refine instance segmentation masks, but not vice-versa.

Formalizing stuff and things segmentation as a single task, Kirillov et al. [15] propose a unified metric called PQ and a strong late fusion baseline combining separate state-of-the-art networks for instance and semantic segmentation. This method uses a simple non-maximum suppression (NMS) heuristic to overlay instance segmentation predictions on top of a "background" of dense semantic segmentation predictions. Saleh et. al [30] show that this heuristic is particularly effective for sim2real transfer of semantic segmentation by first decoupling things and stuff before late fusion. Indeed, stuff classes can have photo-realistic synthetic textures (ensuring stuff segmentation transfer), while objects typically have realistic shapes (ensuring detection-based instance segmentation generalization). This approach leverages the specificity of things and stuff, but not their relation and does not tackle the joint panoptic task. Li et al. [18] propose an end-to-end approach that tackles the unified panoptic problem by reducing it to a semantic segmentation partitioning problem using a fixed object detector and "dummy detections" to capture stuff categories. Their work focuses on the flexibility to handle weak supervision, at the expense of accuracy, yielding significantly worse performance than the panoptic baseline of [15], even when fully supervised.

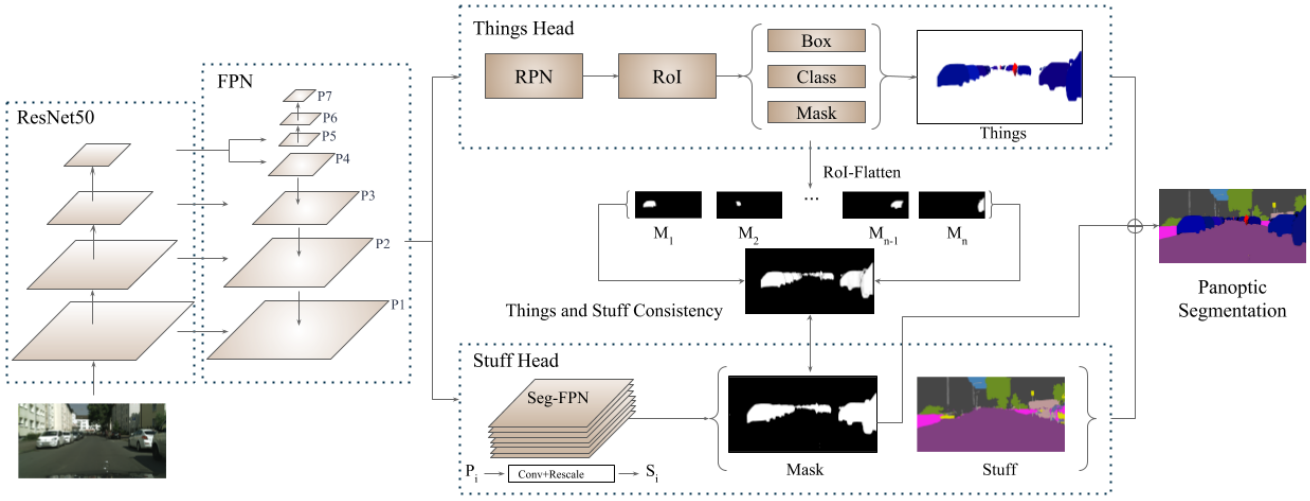


Figure 2: TASCNet: Our unified architecture jointly predicts things, stuff, and a fusion mask. The proposed heads are built on top of a ResNet + FPN backbone. The Stuff Head uses fully convolutional layers to densely predict all stuff classes and an additional things mask. The Things Head uses region-based CNN layers for instance detection and segmentation. In between these two prediction heads, we propose a Things and Stuff Consistency loss to ensure alignment between the predictions.

3. End-to-end Panoptic Segmentation

3.1. TASCNet Architecture

High-performance models for instance and semantic segmentation share similar structures, typically employing deep backbones that generate rich feature representations on top of which task-specific heads are attached [6][12]. Our TASCNet architecture follows this general motif, as is depicted in Figure 2. We use a ResNet50 [13] with an FPN [20] as our backbone, with two task specific heads that share feature maps from the FPN. While the ResNet alone has a large receptive field due to aggressive downsampling, this comes at the expense of spatial resolution and the ability to accurately localize small and large objects. Using an FPN enables us to capture low-level features from deeper within the backbone network to recognize a broader range of object scales with far fewer parameters than dilated convolutions. This is a crucial design choice when considering hardware constraints for the *already* memory-intensive semantic and instance segmentation tasks, let alone the joint learning task [20].

3.1.1 Stuff Head

Taking inspiration from Kirillov et al [2], we leverage the multi-scale features from the FPN with minimal additional parameters to make dense semantic predictions. From each feature map level of the FPN, we:

1. apply a set of 3x3 convolutions, reducing the number of channels from 256 to 128;
2. normalize the layer using group normalization with 16 groups [38];
3. apply an additional set of 3x3 convolutions, maintaining the number of channels;
4. normalize and upsample to the largest FPN feature map size (4x downsampled from the input resolution).

Each output layer is then stacked and one final convolution is applied to predict the class per pixel.

3.1.2 Things Head

To regress instances, we use Region-based CNN heads on top of the FPN, similarly to Mask R-CNN [12]. We augment the FPN with two additional high level context feature maps, similarly to [19]. Improving the instance segmentation architecture was not the primary focus of this work, and we use the same head structures as in [12]. We train the bounding box regression head, class prediction head, and mask head in an end-to-end fashion.

3.2. Things and Stuff Consistency (TASC)

Although the sub-task heads are trained using shared features, the output distributions of the two heads can still drift apart. There are several potential causes of this drift, such as minor differences in annotations for instance vs. semantic segmentations, sub-optimal loss functions that capture separate objectives, and local minima for the sub-tasks that do not optimize the joint criterion.

For panoptic segmentation, however, we aim to train towards a global minimum in which the things and stuff segmentations from the two tasks are identical. We seek to enforce such a shared representation through an intermediate confidence mask reflecting which pixels each task considers to be things vs. stuff.

This mask can be constructed in a differentiable manner from both instance and semantic segmentation outputs. Doing so from dense semantic predictions is trivial. First, we apply a threshold of 0.5 to the logits of the Stuff head. Then, all remaining pixels predicting things classes are assigned to their logit values, and all pixels predicting stuff classes are assigned to 0.

For the Things head, constructing the confidence mask is slightly more involved. At train time, an Region Proposal Network (RPN) proposes RoIs, which are pooled to a fixed size using an RoI-Align operation. The mask head then produces a per-class foreground/background confidence mask for each positive proposal from the RPN. We can then reassemble the global binary mask using a differentiable operation we dub "RoI-Flatten":

1. For each image, we construct an empty tensor of equivalent size to the input image.
2. For each RoI, we only consider the foreground/background mask for the class of the ground truth instance the RoI was assigned to regress.
3. We interpolate each of these single-instance masks, (M_1, ..., M_N), to the size of its corresponding RoI in the input image.
4. A threshold of 0.5 is applied to each mask. The thresholded mask is then added to the RoI's original position in the empty tensor.
5. To obtain our final confidence mask, we normalize by the instance count at each pixel post-threshold.

To encourage our instance and semantic segmentation heads to agree on which pixels are things and which are stuff, we minimize the residual between these two masks using an L_2 loss. This residual is visualized in Figure 3.

3.3. Mask-Guided Fusion

Our learning objective encourages the two masks to agree. Therefore, in a converged TASCNet we can use the semantic segmentation mask to select which pixels are obtained from the instance segmentation output and which pixels are obtained from the semantic segmentation output.

We consequently define a simple post-processing procedure: we add regressed instances into the final panoptic output in decreasing order of confidence, only adding an instance to the output if it has an IoU of under 0.4 with instances that have already been added and an IoU of greater than 0.7 with the mask.

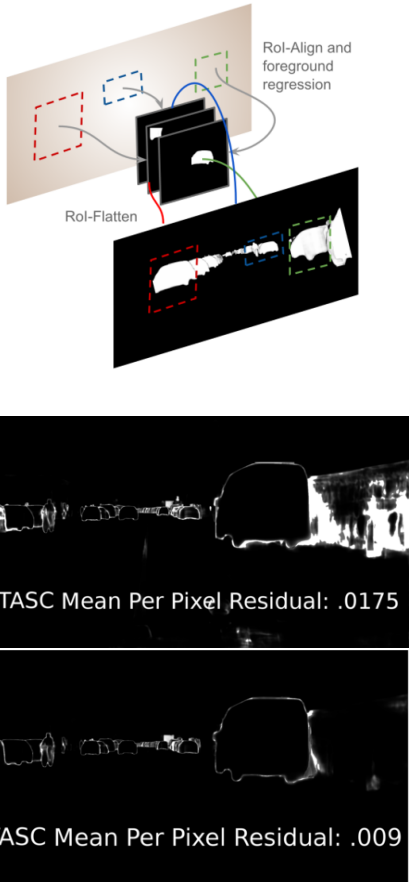


Figure 3: Our proposed "RoI-Flatten" operation (top), and residual error masks without and with cross-task consistency (bottom left and right respectively). We compute the mean per pixel residual over the validation set of Cityscapes and observe a significant reduction when our cross-task consistency is applied.

4. Experiments

4.1. Datasets

We evaluate our proposed approach using two benchmark datasets, Cityscapes [9] and Mapillary Vistas [25]. While these datasets have similar properties in terms of content, there is a large complexity gap between the two.

Cityscapes is comprised of street imagery from Europe and has a total of 5000 densely annotated images with an ontology of 19 classes, 8 thing classes and 11 stuff classes. All the images are at 1024 x 2048 resolution, and are split into separate training, validation, and test sets. We train our model on the "fine" annotations in the training set and test on the provided validation set.

The Mapillary Vistas, while also a street scene dataset, is far more challenging. It consists of a wide variety of geographic settings, camera types, weather conditions, image aspect ratios, and object frequencies. The average resolu-

Method	PQ All	PQ Things	PQ Stuff	AP	AP50	mIoU Stuff	mIoU
Cityscapes Validation Set							
Inplace-ABN, ResNeXt101 [3]		-	-	-	-	-	77.6
Inplace-ABN, WiderResNet38 [3]	-	-	-	-	-	-	**79.4**
Mask R-CNN + PSPNet [15]	**61.2**	54	66.4	36.4	-	**80.6**	-
CRF + PSPNet (Fully Supervised) [18]	53.8	42.5	62.1	-	-	70.4	71.6
TASCNet (Full Ontology)	59.2	56.0	61.5	37.6	64.3	78.2	77.8
TASCNet (Collapsed Ontology)	58.6	54.5	61.5	37.7	64.0	-	-
TASCNet (Full Ontology, Multi-Scale + Flip)	60.4	**56.1**	63.3	**39.0**	**66.8**	78.7	78.0
Mapillary Vistas Validation Set							
Mask R-CNN + FPN Segmentation	32.6	31.8	33.7	20.2	36.5	**57.4**	**47.9**
TASCNet (Full Ontology)	30.7	28.6	33.4	16.6	32.1	57.2	46.9
TASCNet (Collapsed Ontology)	32.6	31.1	**34.4**	18.5	35.0	-	-
TASCNet (Collapsed Ontology, Multi-Scale + Flip)	**34.3**	**34.8**	33.6	**20.4**	**38.0**	-	-

Table 1: **Comparison to the state of the art on Cityscapes and Mapillary Vistas** Our joint network with smaller encoder backbone (ResNet-50) is comparable to other state of the art techniques using separate models. "Collapsed Ontology" here indicates dense semantic prediction of all stuff classes and a single global things class. mIoU performance for panoptic networks is evaluated from panoptic segmentation results collapsed back to dense segmentation predictions.

tion of the images is around 9 megapixels, which causes considerable complications for training deep nets with limited memory. The dataset consists of 18000 training images, 2000 validation images, and 5000 testing images. Annotations for pixel-wise semantic segmentation and instance segmentation are available for the training and validation sets. The labels are defined on an ontology of 65 semantic classes, including 37 thing classes and 28 stuff classes.

Evaluation Metrics

We evaluate our model's performance on Panoptic Segmentation task using the PQ metric proposed by [15],

$$PQ = \frac{\Sigma_{(p,g)\in TP} IoU_{(p,g)}}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \quad (1)$$

where *p* and *g* are matched predicted and ground truth segments exceeding an IoU threshold of 0.5, and TP, FP, FN denote true positives, false positives, and false negatives, respectively.

To better explore the capability of our proposed approach, we also evaluate our model on the separate instance and semantic segmentation tasks. For semantic segmentation, we use the standard metric, Intersection over Union (IoU). For instance segmentation, following recent literature, we average over the AP^r [11] with acceptance IoU from 0.5 to 0.95 in increments of 0.05. We call this metric AP in the rest of the paper without ambiguity. For minor scores, we also report the $AP50 = AP^r(IoU > 0.5)$.

4.2. Experimental Results

We compare our approach with other state-of-the-art panoptic, instance, and semantic segmentation models on the Cityscapes and Vistas datasets. In Table 1 we present this comparison over the relevant metrics for each task. Qualitative examples are also given in Figure 5 All experiments are carried out using a ResNet50 backbone.

We compare to the challenging baseline proposed in [15], which combines outputs from state-of-the-art semantic (PSPNet [41]) and instance segmentation (Mask R-CNN [12]) models. The panoptic segmentation performance of our proposed TASCNet, using only a ResNet-50 backbone and basic test-time augmentation, matches the PQ reported in [15]. We also compare to Li et. al. [18], and the panoptic segmentation performance of our unified TASCNet outperforms their fully supervised joint solution by a large margin. It is also worth noting that when we generate semantic segmentation from our panoptic segmentation outputs we achieve IoU comparable to state-of-the-art semantic segmentation methods using very large backbones, such as [3].

We also note that although the Stuff and Things Heads are trained using the same data augmentation (described in Section 4.3), they respond quite differently to input resolution at test-time. The Stuff Head is very sensitive to input resolution and tends to perform best at resolutions comparable to those used at train-time. The Things Head, conversely, performs well at a broader range of scales and performs best at resolutions higher than those used at train-time. This is likely due to the scale-invariance of the RoI-pooling layers in the Things Head. As a result, for our test-time augmentation numbers we apply multi-scale augmentation for each task, using higher resolutions to obtain the instance segmentation output.

A limited number of panoptic segmentation results on

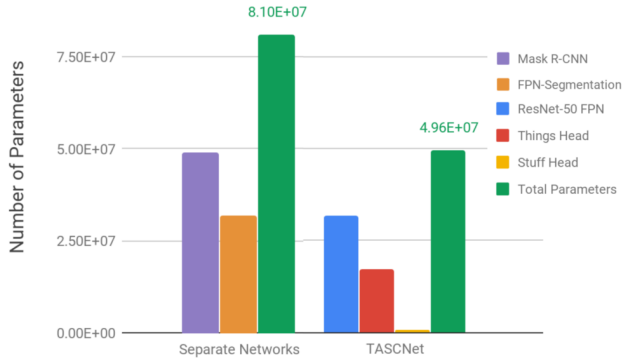


Figure 4: Network parameter efficiency. This graph provides the number of parameters of different models and modules in our experiment. To fulfill the task of panoptic segmentation, our model requires only 61% of parameters compared to separate models using a similar backbone.

Mapillary Vistas has been reported in the literature. [15] reports a reference PQ of 38.3 on a subset of the Vistas test set, combining the winning entries from the LSUN'17 Segmentation Challenge [41, 22]. Megvii report their ECCV'18 Panoptic Segmentation Challenge winning entry in [4]. However, it is difficult to make a fair comparison to their models without additional technical details. Instead, we take our own separate instance and semantic segmentation modules and generate our panoptic segmentation baseline as in [15]. We also present our network performance under different settings.

4.3. Implementation Details

All models were trained using synchronous distributed training over 8 Nvidia V100 GPUs. We fix the batch size of all experiments to 1 per GPU, and correspondingly freeze the BatchNorm layers in the backbone architectures to prevent the layer statistics from dramatically changing. We find that freezing the BatchNorm layers does not affect our performance when compared to networks retrained with unfrozen BatchNorms and larger batch sizes. This strategy enables us to train at very high resolutions, which we find to be critically important for achieving good results.

Unified Data Augmentation

For each input image we apply only one form of data augmentation: an aspect ratio-preserving scale with jitter and randomized left-right flip (similar to that employed in [12] for instance segmentation). In this way, on top of our unified architecture, we also present a unified approach to data augmentation for instance and semantic segmentation. This enables training without the need for two forms of data augmentation (one tailored to instance segmentation and another tailored to semantic segmentation). Conventionally semantic segmentation networks have been trained using crops, but we find that we can achieve equal or better performance with scale jitter alone. Our decision to use a relatively smaller backbone (ResNet50) allows us to train on high resolutions, sometimes even super-resolved images, which enables us to catch finer details and features. The FPN structure also helps to capture some of the lower level features normally emphasized by cropping strategies.

Specifically, for Cityscapes experiments, we jitter the scale from 800px-1400px on the shortest side, maintaining a fixed aspect ratio. We also apply random horizontal flips.

For the Vistas dataset, since the resolution varies so widely, we pick an intermediate scale that maximizes the memory utilization. Hardware memory limits our input resolution to a maximum input scale of 1650px for our full ontology experiments. When we use a reduced ontology, predicting just stuff classes and a binary collapsed "things class with our Stuff Head, we can train with a maximum scale of 2500px.

Optimization Scheme

We use a straightforward learning rate policy similar to [12]. We ramp up the learning rate linearly over the first 500 iterations, starting at 1/3 of the initial base learning rate. We then take two steps in the learning rate over the course of the training run, dividing by 10 each time. For optimization we use stochastic gradient descent with momentum 0.9 and weight decay 10^{-4}. For all experiments, we use learning rates between 0.005 and 0.02.

Test-time augmentation

For both instance and semantic segmentation, we run inference at multiple scales, each with and without a horizontal flip. For regressed bounding boxes, we apply NMS with a cutoff threshold of 0.3 and run the mask head using the resulting set of boxes as proposals. For semantic segmentation, we interpolate the predicted logits for each scale to the raw image resolution, and average logits across scales.

4.4. Ablative Analysis

In this section, we present a thorough ablation study of our proposed method on the Cityscapes dataset. Experimental results are given in Table 2 and Table 3. All tasks, whether separate segmentation tasks or the joint task, are trained with the same ResNet50 + FPN backbone. In all experimental settings, we perform hyper-parameter sweeps to maximize performance so that comparisons are fair.

We first train a set of semantic segmentation models to understand the effect of reducing ontology complexity on the IoU for stuff classes. In particular, we predict three types of ontologies: a full ontology composed of all things

Method	PQ All	PQ Things	PQ Stuff	AP	AP50	mIoU Stuff	mIoU
Separate Models							
Full Ontology Semantic Segmentation	-	-	**61.6**	-	-	77.6	75.1
Stuff Only Semantic Segmentation	-	-	-	-	-	75.6	-
Stuff and Binary Things Semantic Segmentation	-	-	61.1	-	-	76.7	-
Instance Segmentation	-	51.2	-	35.2	61.2	-	-
Fused Full Ontology and Instance Segmentation	57.3	51.5	61.5	35.2	61.2	77.6	75.6
Unified TASCNet Model							
Stuff and Binary Things, w/o TASC	56.0	51.6	59.1	35.4	60.7	-	-
Stuff and Binary Things, w/ TASC	58.6	54.5	61.5	**37.7**	64.0	-	-
Stuff and Binary Things w/ TASC + Mask Fusion	58.9	55.2	61.5	**37.7**	64.0		
Full Ontology, w/o TASC	58.5	54.8	61.3	37.3	63.1	77.4	76.5
Full Ontology, w/ TASC	59.0	55.6	61.5	37.6	**64.3**	**78.2**	**77.8**
Full Ontology, w/ TASC + Mask Fusion	**59.2**	**56**	61.5	37.6	**64.3**	**78.2**	**77.8**

Table 2: **Ablative analysis on Cityscapes** We report panoptic segmentation performance (PQ), instance segmentation performance (Mask AP/AP50), and semantic segmentation performance (mIoU). The same ResNet-50 + FPN backbone is used for all experiments. All results are reported without test time augmentation or model ensembling. mIoU performance for panoptic networks is evaluated from panoptic segmentation results collapsed back to dense segmentation predictions. All models are pretrained on COCO detection.

and stuff, a stuff and binary ontology, in which all things classes are collapsed into a single class, and a stuff class only ontology, which ignores all things pixels. We find that reducing the number of classes has a minor negative effect on learning for the Cityscapes dataset, but can provide valuable memory savings in the multitask setting.

We additionally train an instance segmentation model, and re-implement the heuristic fusion strategy from [15]. Our baselines achieve similar performance to [15] and we take these results as targets for the joint architecture.

We then train our joint architecture to predict each of the three aforementioned ontologies in addition to instance segmentation, and fuse the results using both the heuristic from [15] and our mask-guided fusion.

We find that TASCNet can significantly improve performance on single tasks compared to single models. Our proposed joint model also achieves better PQ performance than the separate models, improving significantly when we explicitly align the output distributions between tasks. This result also furhter demonstrated our outstanding network efficiency as a single network as depicted in Figure 4.

For the second part of the ablation study, we explore the training protocol for our joint network. We pretrain on various datasets and also examine stage-wise training, as shown in Table 3. In single-stage training, we train our joint network from a pretrained backbone and random initialized heads. As expected, we observe that the backbone pretrained on COCO helps the network converge to a better minimum than the ImageNet pretrained one. Freeing the pretrained backbone will also help improve performance. In stage-wise training, we first train the network on a single task (backbone + single head) to converge and then fine-tune the whole network with another head and TASC added. In general, TASCNet is not very sensitive to the different training methods, but joint training without fully trained head tends to converge to better minima. We present more detailed results in the Appendix.

Pretraining	PQ All	PQ Things	PQ Stuff
ImageNet	55.9	50.5	59.8
COCO (Frozen Backbone)	53.3	50.6	55.3
COCO	**59.2**	**56.0**	**61.5**
Stuff Head First	57.3	51.6	61.5
Things Head First	58.5	55.0	61.0

Table 3: **TASCNet Pretraining Protocol on Cityscapes** We pretrain our network on different datasets and measure the effect on PQ. We train the backbone on Imagenet, and subsequently on COCO. We finetune our model in a stage-wise and joint fashion. We also freeze the entire backbone to measure the quality of COCO pretrained features.

5. Conclusion

In this work, we proposed an end-to-end network, TASCNet, to jointly predict stuff and things. We demonstrate that a novel cross-task constraint can boost the performance on instance, semantic, and panoptic segmentation. Using a unified network, our proposed approach is competitive with state-of-the-art models on both the panoptic segmentation task and individual instance and semantic segmentation tasks on several benchmarks, while using far fewer parameters.

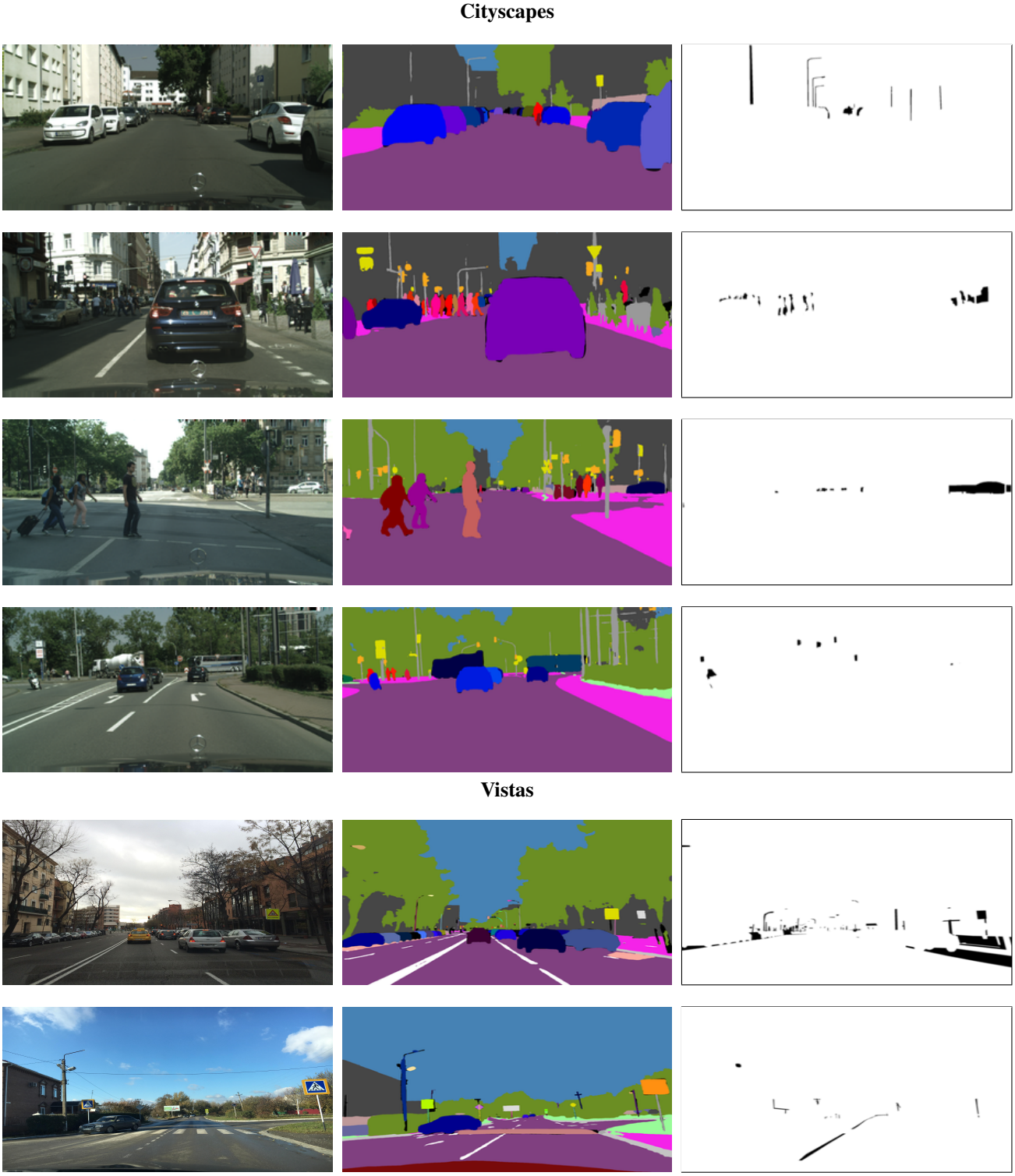


Figure 5: **Panoptic segmentation examples from Cityscapes and Mapillary Vistas.** In panoptic segmentation results, different instances are color-coded with different colors with small variations from the base color of their semantic class. In mismatched segments, segments belongs to true positives are marked as white, while false positive and false negative segments are marked as black.

References

[1] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Human vision and electronic imaging VI*, volume 4299, pages 1–13. International Society for Optics and Photonics, 2001. 1

[2] K. Alexander, H. Kaiming, G. Ross, and P. Dollr. A Unified Architecture for Instance and Semantic Segmentation. `http://presentations.cocodataset.org/COCO17-Stuff-FAIR.pdf`, 2017. Online. 3

[3] S. R. Bulò, L. Porzi, and P. Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. *CoRR, abs/1712.02616, December*, 5, 2017. 1, 2, 5

[4] P. Chao, W. Jingbo, Y. Changqian, L. Xu, L. Huanyu, L. Zeming, Z. Yueqing, Z. Xiangyu, Y. Gang, and S. Jian. Mscoco & mapillary panoptic segmentation challenge 2018. 6

[5] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. *arXiv preprint arXiv:1712.04837*, 2017. 2

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 2, 3

[7] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*, 2017. 2

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 4

[10] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3992–4000, 2015. 2

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 5

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 2, 3, 5, 6

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[14] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[15] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, 2018. 1, 2, 5, 6, 7

[16] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, volume 2, page 8, 2017. 2

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[18] Q. Li, A. Arnab, and P. H. Torr. Weakly- and semi-supervised panoptic segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 5

[19] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 3

[20] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 2, 3

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[22] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Lsun17: Instance segmentation task, ucenter winner team. 6, 11

[23] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 1, 2

[24] R. Mottaghi, X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, June 2014. 2

[25] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 1, 2, 4

[26] D. Neven, B. De Brabandere, S. Georgoulis, and L. Van Gool. Fast scene understanding for autonomous driving. In *Deep Learning for Vehicle Perception, workshop at the IEEE Symposium on Intelligent Vehicles*, pages 1–7, 2017. 1, 2

[27] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015. 1, 2

[28] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 11

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[30] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez. Effective use of synthetic data for urban scene semantic segmentation. *arXiv preprint arXiv:1807.06132*, 2018. 2

[31] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. *arXiv preprint arXiv:1810.04650*, 2018. 2

[32] M. Sun, B.-s. Kim, P. Kohli, and S. Savarese. Relating things and stuff via objectproperty interactions. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1370–1383, 2014. 2

[33] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018. 2

[34] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3001–3008, 2013. 2

[35] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3748–3755, 2014. 2

[36] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005. 2

[37] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2016. 2

[38] Y. Wu and K. He. Group normalization. *CoRR*, abs/1803.08494, 2018. 3

[39] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 702–709. IEEE, 2012. 2

[40] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 1, 2

[41] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 1, 2, 5, 6

[42] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 4. IEEE, 2017. 1

Appendix: Detailed Experimental Results

In this appendix, we provide PQ performance in detail on both Cityscapes (Table 4) and Mapillary Vistas (Table 5) as well as more qualitative results (Figure 6 and Figure 7). These results were obtained using the test-time augmentation described in the main paper (multi-scale and flip).

We note that, in Cityscapes, some of the stuff classes that achieve the lowest PQ scores (e.g. wall and fence) actually perform respectably on IoU. This is because PQ treats *all pixels* from each stuff class as a *single* segment.

This effectively penalizes stuff segmentation more harshly than things segmentation (where there are potentially multiple segments per class). We believe it's important to improve on the PQ metric to strike a better balance between how thing and stuff classes are evaluated.

We can tell from Table 5 that our main challenge on Mapillary Vistas is poor performance on rare classes, even though we used a weighted, bootstrapped loss [28] for training. Similar issues have been observed by existing approaches to semantic segmentation on this dataset [22]. In future work we hope to explore other techniques for balanced batch sampling and rare class bootstrapping to improve TASCNet performance.

The examples in Figure 6 and Figure 7 further depict how our proposed method is able to achieve high quality panoptic segmentation using a unified architecture with a small backbone.

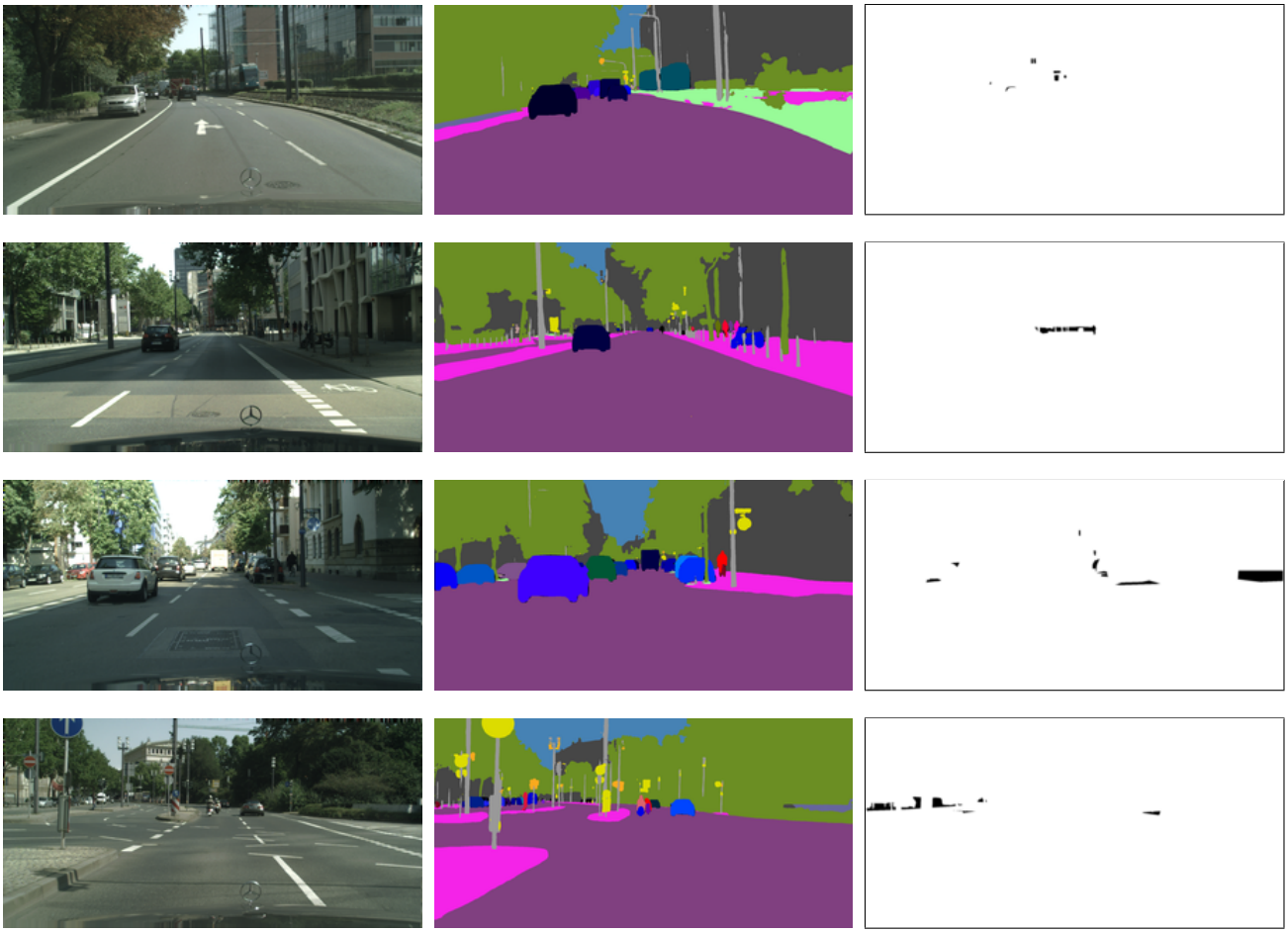
Some examples also further illustrate our discussion of stuff bias in PQ. In the third sample in Figure 7, although we correctly classify a substantial fraction of lane marking pixels, the lane marking PQ for this image is 0 as the fraction is under 50%.

Class	**PQ**	**SQ**	**RQ**	**IoU**
mean	60.4	80.3	73.8	78.0
sidewalk	75.3	83.8	89.9	85.1
building	87.8	89.4	98.2	92.5
wall	25.5	71.2	35.8	55.8
fence	27.0	71.5	37.7	57.8
pole	48.3	64.6	74.8	64.3
traffic-light	44.7	68.6	65.2	71.2
traffic-sign	66.1	75.9	87.0	78.7
vegetation	88.6	90.3	98.2	92.5
terrain	28.9	73.4	39.3	64.4
sky	86.2	92.3	93.4	95.1
person	51.6	76.9	67.0	83.0
rider	53.5	71.7	74.6	68.0
car	64.8	83.2	77.8	93.6
truck	53.3	86.3	61.7	65.5
bus	72.9	88.5	82.4	87.7
train	65.6	83.1	78.9	83.5
motorcycle	44.5	72.3	61.5	68.4
bicycle	42.2	71.6	58.9	76.0

Table 4: Panoptic Quality (PQ) on Cityscapes. IoUs are also included. Note that PQ, particularly for stuff classes, is a very stringent metric.

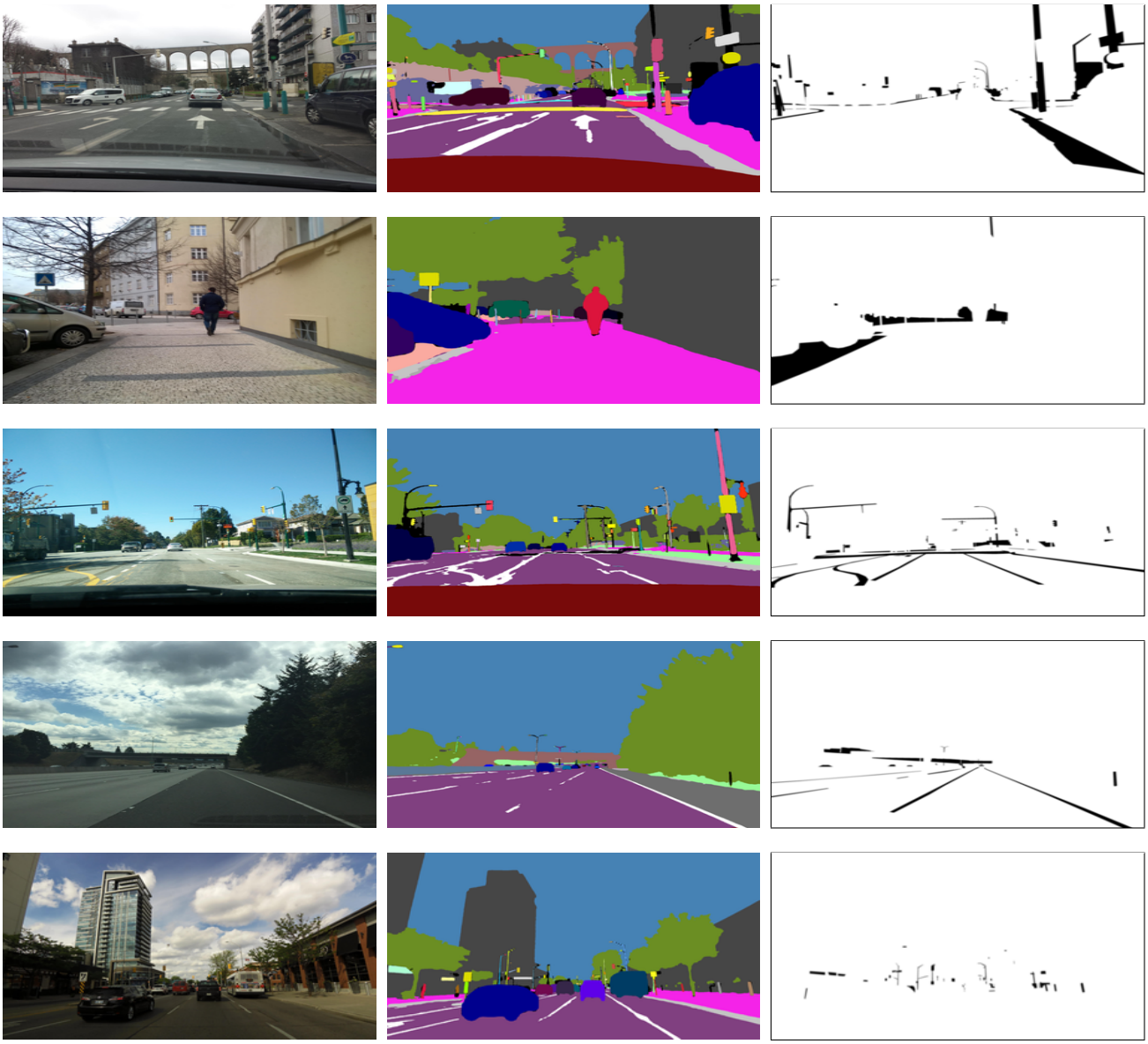
Class	PQ	SQ	RQ	Class	PQ	SQ	RQ
mean	34.3	74	43.5	object–banner	30.6	82.3	37.1
animal–bird	0.0	0.0	0.0	object–bench	26.8	73.8	36.4
animal–ground-animal	39.1	80.5	48.5	object–bike-rack	6.7	71.2	9.4
construction–barrier–curb	42.9	70.5	60.9	object–billboard	40.1	82.1	48.8
construction–barrier–fence	32.0	72.2	44.3	object–catch-basin	35.7	75.4	47.4
construction–barrier–guard-rail	29.5	73.2	40.3	object–cctv-camera	21.0	69.5	30.2
construction–barrier–other-barrier	24.3	77.4	31.3	object–fire-hydrant	56.8	80.8	70.2
construction–barrier–wall	19.4	72.7	26.7	object–junction-box	41.1	85.4	48.1
construction–flat–bike-lane	13.0	69.9	18.7	object–mailbox	23.0	79.0	29.2
construction–flat–crosswalk-plain	32.9	75.3	43.7	object–manhole	47.7	81.2	58.8
construction–flat–curb-cut	3.3	62.0	5.4	object–phone-booth	15.1	84.7	17.9
construction–flat–parking	5.9	64.2	9.2	object–pothole	0.6	63.0	1.0
construction–flat–pedestrian-area	19.1	85.2	22.4	object–street-light	46.0	74.0	62.1
construction–flat–rail-track	11.9	70.5	16.9	object–support–pole	33.8	70.3	48.0
construction–flat–road	83.1	89.1	93.3	object–support–traffic-sign-frame	18.8	64.4	29.2
construction–flat–service-lane	28.6	80.2	35.7	object–support–utility-pole	36.9	68.4	53.9
construction–flat–sidewalk	52.4	77.6	67.6	object–traffic-light	59.0	79.4	74.4
construction–structure–bridge	33.1	76.6	43.2	object–traffic-sign–back	39.0	74.9	52.2
construction–structure–building	69.2	83.0	83.5	object–traffic-sign–front	58.8	83.8	70.2
construction–structure–tunnel	8.9	56.8	15.7	object–trash-can	48.5	83.7	57.9
human–person	54.7	78.7	69.5	object–vehicle–bicycle	37.7	72.1	52.2
human–rider–bicyclist	43.3	73.6	58.8	object–vehicle–boat	17.1	67.8	25.3
human–rider–motorcyclist	38.2	70.3	54.3	object–vehicle–bus	54.6	87.6	62.3
human–rider–other-rider	7.8	52.8	14.8	object–vehicle–car	68.7	85.8	80.1
marking–crosswalk-zebra	46.1	76.1	60.5	object–vehicle–caravan	0.0	0.0	0.0
marking–general	41.1	68.6	59.9	object–vehicle–motorcycle	45.0	73.1	61.6
nature–mountain	22.1	71.8	30.8	object–vehicle–on-rails	6.9	79.4	8.7
nature–sand	4.2	81.5	5.1	object–vehicle–other-vehicle	22.2	74.6	29.7
nature–sky	96.0	96.8	99.2	object–vehicle–trailer	13.6	79.3	17.1
nature–snow	33.8	82.3	41.1	object–vehicle–truck	51.6	86.3	59.8
nature–terrain	37.8	76.9	49.2	object–vehicle–wheeled-slow	27.8	70.3	39.5
nature–vegetation	81.3	86.0	94.5	void–car-mount	51.7	85.2	60.7
nature–water	17.7	75.4	23.4	void–ego-vehicle	71.5	90.4	79.2

Table 5: Panoptic Quality (PQ) on Mapillary Vistas.



Raw Image | TASCNet Panoptic Segmentation | Mismatched Segments

Figure 6: **Panoptic Segmentation Examples from Cityscapes.** In panoptic segmentation results, different instances are color-coded with different colors with small variations from the base color of their semantic class. In mismatched segments, segments belongs to true positives are marked as white, while false positive and false negative segments are marked as black.



Raw Image | TASCNet Panoptic Segmentation | Mismatched Segments

Figure 7: **Panoptic Segmentation Examples from Mapillary Vistas.** In panoptic segmentation results, different instances are color-coded with different colors with small variations from the base color of their semantic class. In mismatched segments, segments belongs to true positives are marked as white, while false positive and false negative segments are marked as black.