Supplementary Material of: Segmenting Known Objects and Unseen Unknowns without Prior Knowledge

Stefano Gasperini[1,2,◦] Alvaro Marcos-Ramiro[2] Michael Schmidt[2]
Nassir Navab[1] Benjamin Busam[1] Federico Tombari[1,3]

[1] Technical University of Munich [2] BMW Group [3] Google

A. Supplementary Material

In this appendix, we include further details and results. Specifically, Sections A.1, A.2 and A.3 provide deeper insights on the proposed setting, the method, and the experimental setup, respectively, while Sections A.4 and A.5 contain more results, both quantitative and qualitative.

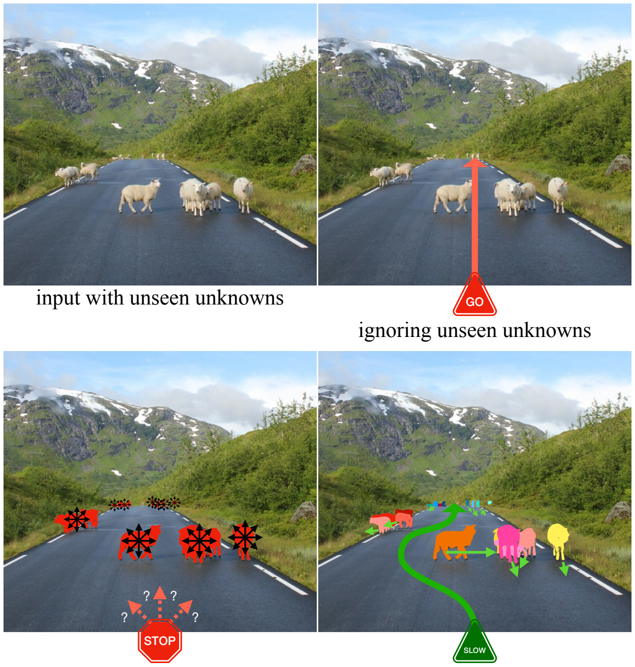
A.1. Additional Details on the Setting

In this section, we describe the benefits of the proposed holistic segmentation setting in greater detail, considering both the impact on downstream tasks and the differences with other perception tasks addressing unknown objects.

The proposed holistic segmentation setting aims to segment any unseen, unknown objects without prior knowledge about the unknowns while segmenting known areas. In this context, "unseen unknowns" means any object of any category outside the known classes learned during training, such as the sheep in Figure 6 for a method trained on, e.g., Cityscapes [7], as well as unidentified and distorted parts following a car accident.

A.1.1 Motivation

The importance of identifying unseen unknowns arises from safety-critical scenarios, such as autonomous driving, where ignoring them can lead to dangerous consequences when simply using the predicted segments for downstream tasks, e.g., path planning. This is shown in the top right of Figure 6. Since even large-scale datasets are limited representations of the real world, there will always be corner cases and long tail samples which are problematic for standard models [15]. Therefore, it is crucial to identify these cases and then deal with them safely via downstream tasks.


input with unseen unknowns

ignoring unseen unknowns

ignoring individual unknowns

with holistic segmentation

Figure 6. Motivation diagram for identifying unseen, unknown objects on a sample of [3] considering path planning as a downstream task, and hypothesizing that *sheep* is not part of the training data (i.e., unseen unknown). Shown segments are not predictions. State-of-the-art approaches dangerously ignore unknowns (top right) [5]. OOD segmentation does not identify instances of unknowns (bottom right) [14], making it difficult for downstream tasks as the unknowns cannot be tracked, and their trajectory cannot be predicted. The proposed setting (bottom right) identifies individual unseen unknowns, leading to a safe path.

OOD segmentation [14], i.e., segmenting unknown areas as a whole and not identifying individual instances of unknowns, flags the presence of something unknown in the input. In the context of downstream tasks, such as path planning, a single OOD segment (bottom left in the figure) could trigger an alert state, leading to a potential stop, which is a

[◦] This work was conducted while working at BMW Group. Contact author: Stefano Gasperini (*stefano.gasperini@tum.de*).

Setting	Data Assumptions	Identifiable Objects	Not Identifiable Objects
open-set panoptic segm. [13, 22]	unknowns are already in the training data, within *void* areas	known and unlabeled objects present in the training data as *void*	categories outside of the training data [13]
open-vocabulary, zero-shot [23, 12]	the underlying language model knows about every unknown	objects known by the language model [19]	categories outside of the training data of the language model [19]
holistic segmentation [ours]	**none**	**any** known and unknown (i.e., **unseen**) object	**none**

Table 4. Comparison of tasks and settings dealing with instances of unknown objects. The second column (Data Assumptions) is related to unknowns. The 2 rightmost columns represent the objects that are theoretically identifiable or not, given the setting.

safe state. However, given that OOD segmentation does not separate unknown objects into instances, once found, it is unclear whether they are moving or static, which means that it would be difficult for path planning. Instead, by segmenting instances of unseen unknowns (bottom right in the figure), holistic segmentation allows tracking unseen objects and estimating their trajectory, leading to a safe path. This motivates the instance segmentation of unknowns, which brings benefits similar to those of instance segmentation compared to semantic segmentation for known objects.

Also critical is the ability to deal with any unseen, unknown object category and not be restricted to a limited subset of them. This is of utmost importance to address the wide variability of objects and scenarios encountered in the real world. While previous settings focused on re-identifying already-seen objects [13, 23], we design holistic segmentation specifically to address any unseen category.

A.1.2 Comparison with Other Settings

As shown in Table 4, compared with other tasks and settings also dealing with unknown objects, the proposed holistic segmentation makes no assumptions about the unknown objects, allowing one to segment any objects. Instead, zero-shot and open-vocabulary approaches assume that text descriptions of unknown objects are available [12, 23]. Open-set panoptic segmentation methods assume unknowns are confined within *void* regions at training and test time [22, 13]. In the latter case, *void* may not be available or not sufficiently large and diverse (as in Cityscapes [7]), depending on the training data. Due to their construction, both of these setups inherently restrict the pool of recognizable objects to those for which text descriptions are available through a vision-language model (open-vocabulary) or to those present within their own training set (open-set panoptic).

For example, for the scene in Figure 6, because of its setup, EOPSN [13] cannot identify any *sheep* unless a vast amount of images containing *sheep* is part of its training data (with *sheep* being labeled as *void*, or directly as a dedicated class *sheep*). Open-vocabulary methods would rely on the fact that a language model [19] already knows about *sheep* to be able to identify them in the image. While the concept of *sheep* is relatively simple and could be assumed to be known by a large language model, there is no guarantee that such a model would know about every possible object and scene that can be encountered in real life (e.g., unidentified pieces on the road following a car accident), meaning that open-vocabulary approaches cannot deal with long tail samples from the distribution of the natural world, simply because their language model cannot process them.

Again, given that datasets include by definition only a fraction of the diversity of the world [15], also datasets to test the ability of a model to identify unknowns are limited [18, 3, 1], containing only a small amount of the possible objects and situations that can be encountered in real life. Therefore, to operate reliably in real unconstrained scenarios, it is of utmost importance not to have limitations on the types of recognizable objects, which should go beyond those found in existing datasets. Instead, relying on a language model to identify unknowns is equivalent to shifting the unknown problem to a different model. As shown with CLIP by Radford et al. [19], large language models also have issues with OOD samples. For example, unidentified broken car parts lying on the ground after an accident would be difficult to describe, so it would be problematic for language models. Thus, when given inputs that are unseen and unknown to the underlying language model, open-vocabulary and zero-shot methods would fail to identify the unknown objects. Furthermore, existing open-set panoptic works rely on the presence of unknowns (intended as unlabeled) directly in the training data through the *void* class. This highlights the need for a new and unconstrained solution.

For these reasons, the critical differences between the proposed holistic segmentation setting and previous tasks are that holistic segmentation is not constrained in terms of the types of unknown objects that are identifiable and that holistic segmentation does not assume the presence of unknowns in the training data, thereby segmenting **any unseen, unknown object without any prior knowledge about unknowns**. Limited by design by either the unknowns that are known to the underlying language model (e.g., open-vocabulary) or the unknowns that are directly present in the training data (e.g., open-set panoptic segmentation), previous tasks do not enable the identification of any instance of unknowns and rely on prior knowledge about unknowns and their data distribution (e.g., through CLIP [19] or by learning *void*).

A.2. Additional Details on the Method

Loss functions As described in Section 4.3, the proposed method is trained with a combination of losses: a semantic loss $\mathcal{L}_s$, an object detection loss $\mathcal{L}_o$, a prototype loss $\mathcal{L}_p$, and a discriminative loss $\mathcal{L}_d$. The discriminative loss is aimed at learning meaningful embeddings. It is composed of three different terms [8], namely variance $\mathcal{L}_{va}$ to attract elements towards the mean, distance $\mathcal{L}_{di}$ to push away different groups, and regularization $\mathcal{L}_{re}$ to prevent the divergence of clusters from the origin:

$$\begin{aligned}
\mathcal{L}_d &= \lambda_{41}\mathcal{L}_{va} + \lambda_{42}\mathcal{L}_{di} + \lambda_{43}\mathcal{L}_{re} \\
\mathcal{L}_{va} &= \tfrac{1}{|\Omega|}\textstyle\sum_{\omega\in\Omega}\tfrac{1}{N_\omega}\sum_{a=1}^{N_\omega}\left[||\mu_\omega-\phi_a||-\delta_v\right]_+^2 \\
\mathcal{L}_{di} &= \tfrac{1}{|\Omega|(|\Omega|-1)}\textstyle\sum_{\omega_A\in\Omega}\sum_{\omega_B\in\Omega}[2\delta_d- \\
&\qquad\qquad ||\mu_{\omega_A}-\mu_{\omega_B}||]_+^2 \\
\mathcal{L}_{re} &= \tfrac{1}{|\Omega|}\textstyle\sum_{\omega\in\Omega}||\mu_\omega||
\end{aligned} \tag{1}$$

where: $|\Omega|$ is the number of prototypes, N_ω is the number of embeddings associated to the prototype ω, μ_ω is the mean embedding of the cluster related to ω, $||\cdot||$ is the L2 distance, $[x]_+ = \max(0,x)$ is the hinge (i.e., until which threshold the terms are active [8]), $\omega_A \neq \omega_B$, and we follow [8] for the hyperparameters, e.g., $\lambda_{41} = \lambda_{42} = 1$ and $\lambda_{43} = 0.001$.

Clustering unseen unknowns As described in Section 4.2, we use DBSCAN [9] to cluster the embeddings of unknown regions into individual unknown objects. Specifically, DBSCAN has multiple advantages: it does not need the number of clusters as input (which is unknown in our case), it is effective and very fast, has a low memory footprint, and distinguishes outliers (Table 3 shows the impact of this feature with A3-A4). Although other traditional clustering methods (e.g., Mean Shift, Affinity Propagation, Birch) are theoretically applicable in our setting, they come with drawbacks (e.g., have high memory requirements, do not output outliers, are significantly slower, or tend to deliver sub-optimal results). On the other hand, popular approaches that require the number of clusters as input cannot be applied in our settings (e.g., K-Means). Hence, DBSCAN was selected.

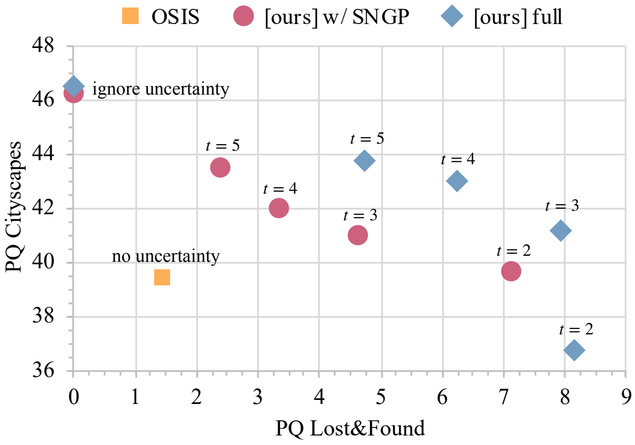


Figure 7. Trade-off between known (i.e., Cityscapes [7] validation set) and unknown (i.e., Lost&Found [18] test set) performance introduced by OSIS [22], compared to our approach, both using SNGP [17] and our improved DPN (i.e., [ours] full, in blue). The different data points are obtained by varying the parameter t.

A.3. Additional Details on the Experimental Setup

Clustering parameters DBSCAN requires two parameters: $minPts$ (number of points in a neighborhood to count as a core point) and ϵ (maximum neighborhood size). To find such parameters, we trained a model (i.e., on Cityscapes [7] or MS COCO [16]), then selected ($minPts$, ϵ) with a simple grid search maximizing PQ on a random subset of the known dataset (i.e., without unknowns). Towards this end, we formed instances as follows: ignoring the detection output (i.e., using only the embeddings) and determining their class via majority voting from the semantic output. In particular, when finding ($minPts$, ϵ), this means treating the embeddings of knowns as if they were unknowns (apart from their semantic class), assuming that the model treats them similarly. It is essential to consider that the parameters were selected on the known objects (i.e., from Cityscapes or COCO), despite DBSCAN being used only for separating unknowns (i.e., in the Lost&Found [18] dataset or the held out classes of COCO). We did this to maintain the unknowns completely unseen (i.e., only as test set), as in real scenarios.

Comparisons with previous works As described in Section 5.1, we compared our uncertainty-based solution with prior works learning the *void* class and tested various uncertainty estimators within our proposed framework. The following paragraphs further detail how these other methods were trained.

Following our setup of training on Cityscapes [7] and transferring to Lost& Found [18] without any fine-tuning, prior works addressing open-set panoptic segmentation

	L&F (*unseen*)		open CS
Uncertainty method	AP	FPR$_{95}$ ↓	mIoU
softmax	16.72	22.88	**71.77**
MC Dropout [10]	11.22	13.94	68.31
DML [2]	3.14	83.04	69.86
DUQ [21]	5.43	26.64	68.78
DPN [20]	5.43	19.79	66.99
SML [14]	16.91	51.67	70.69
SNGP [17]	22.70	**12.02**	70.68
improved DPN [ours]	**25.44**	19.10	70.10

Table 5. Comparison of open-set semantic segmentation on Lost&Found [18] test set of uncertainty estimators based on DeepLabV3+ [4] and trained only on Cityscapes (CS) [7].

(i.e., OSIS [22] and EOSPN [13]) were trained by learning the *void* class of Cityscapes, unlike our U3HS. This unlabeled class comprises all pixels that do not fulfill the requirements to be part of one of the standard 19 annotated classes. Some of these *void* pixels are systematic, e.g., the back side of traffic signs and street lights (excluding poles). By exploiting the variability within *void*, the models learn the extra class decision boundary as a fallback covering anything far from the other classes. To do so, OSIS learns a constant U representing such boundary. In particular, we adapted OSIS from LiDAR point clouds to RGB images, applying it to each pixel instead of point and changing architecture accordingly. As for ours and all other models in this work, we used a ResNet50 [11] as backbone and decoders following the structure of DeepLabV3+ [4]. Moreover, to keep the GPU memory low, we used the same $F = 8$ for the embedding size as in our U3HS. For the experiments on MS COCO [16], we followed the K=5% setup of EOPSN [13], turning 4 classes into *void* to facilitate prior works learning on *void*, by ensuring a diverse distribution of its pixels, as they now cover a diverse set of classes (e.g., *pizza* and *car*).

For the other uncertainty estimation approaches evaluated in this work, we used the authors descriptions and implementations, adapting [21, 17] from image classification to semantic and panoptic segmentation. For DML [2], we used the authors best hyperparameters, therefore a variance loss weight $\gamma_{VL} = 0.01$, and weights $\beta = 20$ and $\gamma = 0.6$. For SML [14], we did not employ the boundary suppression, as it did not improve the results. This might be due to Lost&Found [18] being annotated only for the OOD objects and a coarse road segment. For DUQ [21], we used an embedding dimension of $m = 8$, due to constrained training resources, same as our $F = 8$. Then, we used length scale $\sigma^2 = 0.3$ and exponential smoothing factor $\gamma = 0.999$. For SNGP [17], we again used an embedding dimension $D = 8$ (due to the limited training resources), no layer norm for the embeddings, an exponential smoothing factor $\gamma = 0.99$ for updating Σ and 50 samples for Monte Carlo averaging to estimate the uncertainty.

Configuration			L&F (*unseen*)		open CS
Ref.	Activ.F.	KL	AP	FPR$_{95}$ ↓	mIoU
[20]	*exp*	yes	5.43	19.89	66.99
[ours]	*softplus*	yes	3.43	25.97	64.36
[ours]	*softplus*	no	**25.44**	**19.10**	**70.10**

Table 6. Ablation study on uncertainty estimates for open-set semantic segmentation. Models trained only on Cityscapes [7].

MS COCO As described in the main paper, given that there is no official set of unknown classes for MS COCO [16], we treat as unknown the least frequent 20% known classes. These classes are: *baseball bat*, *bear*, *fire hydrant*, *frisbee*, *hair drier*, *hot dog*, *keyboard*, *microwave*, *mouse*, *parking meter*, *refrigerator*, *scissors*, *snowboard*, *stop sign*, *toaster*, and *toothbrush*. We held out all training samples where any of these 16 classes appeared, such that they were completely unseen to the models.

Ablation study for holistic segmentation With reference to Table 3, A1 is our baseline, which was built upon OSIS [22]. As OSIS, A1 included learned instance-aware embeddings, but unlike OSIS, it featured a semantic decoder delivering semantic segmentation and uncertainty estimates based on the semantic output via our improved DPN. Moreover, as for all our models, A1 did not learn the *void* class (unlike OSIS). A2 featured the relaxed score for the embedding association (described in Section 4.1), which lets the variance be indirectly controlled by the final task (i.e., the loss $\mathcal{L}_p$, Section 4.3). Unlike A1 and A2, which had a shared head between embeddings and prototypes (i.e., as in OSIS), A3 introduced a dedicated prototype head. In practice, this meant having more layers fully dedicated to the embeddings and the prototypes separately instead of sharing the computation until a later stage. Therefore, this allowed for more expressive and purposed features. A4 did not reassign to the known classes the outliers obtained from clustering unknowns via DBSCAN. Therefore, these pixels were kept unknown and shared the same instance ID. A5 did not perform majority voting (Section 4.1). This meant directly assigning the semantic classes predicted by the semantic branch to all known instance pixels instead of enforcing coherence within an instance. This caused the instances to be fragmented according to how many semantic classes they contained, decreasing RQ. Finally, A6 predicted the semantic classes for *stuff* areas directly from the semantic prediction branch instead of matching the embeddings with *stuff* prototypes as in A1-A5 (Section 4.1).

A.4. Additional Quantitative Results

Trade-off between known and unknown Figure 7 shows the trade-off between the performance on known and

ResNet depth	F	PQ	RQ	SQ
18	2	33.0	42.3	77.9
18	4	38.9	49.8	78.0
18	8	41.3	52.7	78.3
18	16	**42.3**	**53.8**	**78.7**
18	32	42.1	53.6	78.5
50	8	**47.7**	**60.4**	**79.0**

Table 7. Different embedding dimensions F on closed-set panoptic segmentation on the validation set of Cityscapes [7]. The first column indicates the depth of the ResNet [11] backbone used (i.e., 18 for ResNet18).

Method	Clustering	PQ	RQ	SQ
U3HS [ours]	Mean Shift	2.71	3.91	**69.22**
U3HS [ours]	DBSCAN	**9.36**	**14.83**	63.14

Table 8. Transfer from Cityscapes [7] to Lost&Found-300 [18] test set (i.e., on the first 300 samples, see Section A.4). DBSCAN (our choice) is compared to Mean Shift to cluster the embeddings of unknown areas.

unknown for our framework, both with SNGP [17] and our improved DPN, compared to that of OSIS [22]. The different data points were extracted by evaluating the outputs at different thresholds t, namely $[2, ..., 5]$, and ignoring the uncertainty estimates entirely (i.e., closed-set, reported where PQ Lost&Found is 0). The hyperparameter t directly affects how high the uncertainty estimates must be for their associated pixels to be considered unknown. This has an impact on the performance on open-set Cityscapes [7] and Lost&Found [18], since changing in output what is considered unknown alters what is regarded as in-domain (i.e., known) as well. OSIS [22] does not have such a hyperparameter as it considers unknown everything predicted as *void*. Overall, it can be seen that our proposed framework offers a better trade-off in both configurations (red and blue) than that of OSIS [22] (yellow). Furthermore, using our full approach (i.e., our framework with our improved DPN) typically gave the best trade-off between known and unknown without compromising the metrics too much (blue).

Unknowns in semantic segmentation In Table 5, we compare the ability of a wide variety of uncertainty estimators (i.e., [20, 21, 17, 14, 2], and MC Dropout with 25 runs [10]) to find unknowns in a semantic setting on Lost&Found [18], after training on Cityscapes [7]. This meant retraining all methods under the same conditions while also extending DPN [20], DUQ [21], and SNGP [17] to semantic segmentation. Semantic models (Tables 5 and 6) used smaller crops sized 512×256 compared to the other experiments. For uncertainty estimation, we evaluated the ability to identify unknowns reporting the AP on the unknown class [18], as well as the false positive rate at the recall 95 (FPR$_{95}$). For semantic segmentation on Cityscapes, we computed the mIoU. As seen in Table 3, DUQ [21] and DPN [20] performed worse than SNGP [17]. MC Dropout [10] underperformed *softmax*, probably due to the contrasting opinions from 25 forward passes. Our method was the best at finding unknowns (AP) with high-quality uncertainty estimates (FPR$_{95}$). Table 5 also reports the mIoU on Cityscapes (CS), showing that all methods introduce a trade-off between OOD and in-domain outputs, as overestimating the uncertainty decreases the in-domain mIoU. Balancing these two complementary aspects is not trivial, with our approach and SNGP managing it best.

Ablation on uncertainty estimation Table 6 compares the DPN [20] we adapted from image classification to semantic segmentation with our extension. Our improvements were oriented to simplify the training process and help convergence. First we applied the *softplus* activation function to the last semantic layer, instead of *exp* as in DPN [20]. We chose *softplus* because it grows slower than *exp* and it is smooth, differentiable everywhere, and monotonic. This significantly improved the training stability at the cost of a reduced quality of the uncertainty estimates. Finally, due to the complexity of modeling the target distribution in our setting, omitting the KL term used by DPN [20] further stabilized training and boosted the performance on all metrics.

Impact of embedding size and architecture Table 7 shows the effect of different embeddings dimensions F on a smaller ResNet18 [11]. In the rest of this work, all experiments used $F = 8$ and ResNet50, as in the last line of the table, due to constrained training resources. The embedding dimension directly affects the learning capability of the model. Since the instance-aware embeddings are a critical part of the output, a smaller F is linked to inexpressive embeddings that cannot be as discriminative as those from a larger F. Therefore, increasing F improved all metrics except for the larger $F = 32$. This can be attributed to the small ResNet18 backbone being already saturated at $F = 16$, unable to extract rich and detailed features for the larger embeddings to exploit. With a larger model (e.g., ResNet101), even higher embedding dimensions F might be beneficial. Table 7 shows that our proposed approach, given less constrained resources, could deliver better results when using an embedding dimension higher than the $F = 8$ employed across this work. The table also shows the comparison between ResNet18 and ResNet50, with the latter delivering over 15% higher PQ at the same $F = 8$. This shows how our proposed approach would perform with a larger backbone.

Impact of the clustering method In Table 8 we compare two popular clustering methods within our U3HS framework, namely DBSCAN and Mean Shift [6]. Due to the very high computation effort and memory required by Mean Shift, we opted for the following setup for this ex-

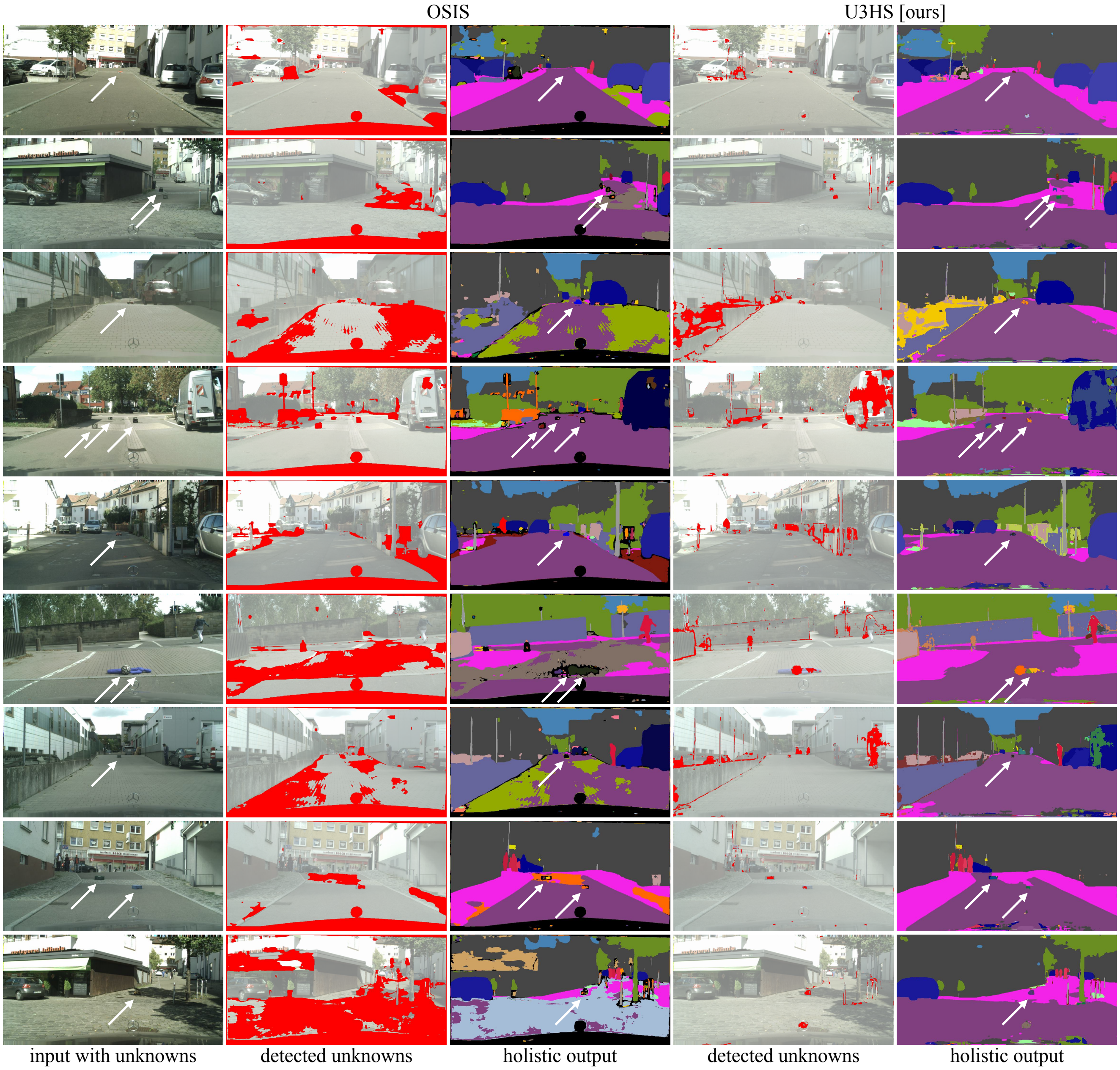


Figure 8. Example predictions of OSIS [22] and the proposed U3HS on unknown categories from the test set of Lost&Found [18]. The models were trained on Cityscapes [7] and transferred to Lost&Found without any fine-tuning. OSIS found unknowns as the *void* class (learned during training), while our U3HS discovered them via uncertainty estimation. Black regions in OSIS's outputs, including around unknowns, represent pixels predicted as part of the unknown instance of the ego vehicle bonnet: since the bonnet is labeled as *void* in the training set, OSIS learned it as such and it turned it into an unknown instance at inference time. White arrows mark labeled OOD objects.

periment. First, instead of a standard CPU implementation, we used a parallelized CUDA version of the algorithm [24]. Then, due to the still very high memory requirements, specific samples of Lost&Found caused memory issues. Therefore, we reduced the size of the test set of Lost&Found [18] to its first 300 samples (Lost&Found-300), which were not problematic. These 300 samples are sufficient to indicate the effect of using Mean Shift instead of DBSCAN. Table 8 shows the superiority of DBSCAN for this setting, with a 3.5x higher PQ and 3.8x better RQ. In particular, RQ should be the focus as we compare instance segmentation of unknowns.

Additional details on EOPSN vs. OSIS As described in Section 5.2, EOPSN always diverged on Cityscapes despite numerous attempts, leading to null true positives, as shown in Table 1. OSIS did not suffer from this issue:

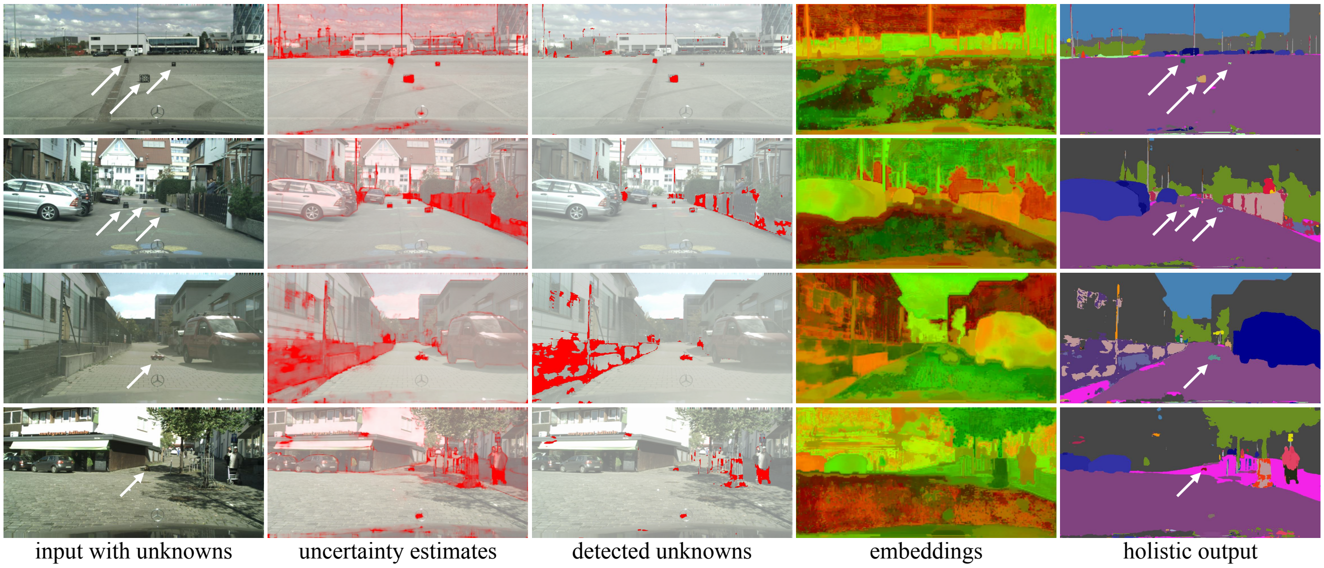


Figure 9. Additional example predictions of the proposed U3HS on unknown categories from the test set of Lost&Found [18]. The model was trained on Cityscapes [7] and transferred to Lost&Found without any fine-tuning. White arrows mark labeled OOD objects.

EOPSN's mining strategy requires associating similar "unknowns" across different inputs, but OSIS operates frame-by-frame. Since *void* pixels are unstructured and undefined in Cityscapes, EOPSN's association fails. As this process is a fundamental step of its training procedure, it makes EOPSN diverge and leads to null scores due to the absence of true positives. Instead, in EOPSN's setup (i.e., re-identifying unlabeled objects seen during training), such associations can be made across the instances of the classes EOPSN's authors treat as *void* or "unknown" (e.g., all cars in their setup, as in Table 2). Therefore, on MS COCO (Table 2), EOPSN managed to identify a few true positives, thereby scoring more than 0 for PQ and RQ and significantly more for SQ as it considers only the IoU of matched segments (TP) and not the wrong predictions (FP and FN).

Varying number of unknowns Both EOPSN and DDOSP showed that their performance drops across the board by increasing the amount of *void* classes to detect (i.e., K, pseudo-unknowns), especially for unknowns. As shown in Table 1, they perform poorly also with $K = 0$. Instead, our U3HS does not rely on *void*, so it is unaffected by K or what is assigned to *void*. This means that U3HS's performance varies only slightly with different Ks, as for random initialization. Moreover, whether turning known classes into *void* (MS COCO, Table 2, with $K = 5\%$) or not (Lost&Found, Table 1, $K = 0$), our method outperforms prior works despite letting the others learn from unknowns via *void*. Figure 8 shows this qualitatively. Thanks to uncertainty estimation, our setup has an edge with unseen categories (e.g., Table 1). Increasing K for open-set works (i.e., treating more classes as *void*) means reducing the number of classes that can be segmented semantically. Therefore, the proposed setting is more practical than open-set panoptic and open-vocabulary because, for ours, no unknowns need to be part of the training of any model (simpler data collection), and ours detects any unseen categories.

A.5. Additional Qualitative Results

A.5.1 Qualitative Comparison

Figure 8 shows a comparison of the predictions of the proposed U3HS with the prior work OSIS [22], as well as the regions each predicted as unknown, on a set of samples from Lost&Found [18]. In particular, ours found unknowns as segments estimated as highly uncertain, and OSIS found them as the pixels predicted to be part of the learned *void* class.

From the images, it can be seen how for the most part, OSIS managed to learn a relatively good class boundary around the *void* class, as it was typically able to predict the OOD objects as unknown via *void*. This is interesting as it shows how OSIS can potentially work with challenging unseen unknowns. However, the same figure also shows the substantial limitations of learning and predicting *void* due to the assumptions about the data distributions this entails. In the first image, OSIS completely ignored the unknown object, assigning it to the *road* class, while in the fifth image, it detected the toy as *car*. In contrast, in the last picture, OSIS predicted almost everything as unknown. This proves how the binary aspect introduced by predicting the *void* class (a pixel is either unknown, by being *void*, or known, if another class) does not cope well with the diversity and unpredictability of the scenes in unconstrained real-world settings. Specifically, predicting the *void* class severely relies on the closed-set training data, as the success of such a method is directly related to the diversity of the *void* class seen during training, which is limited as it cannot correctly sample the long tail of the data distribution [15].

Nevertheless, as shown already in Section 5, estimating the uncertainty allows to properly cope with unknown objects by adding an extra layer of prediction. Contrary to the idea of prior works (Section 2) of predicting unknowns via the *void* class, which directly competes with the other semantic classes for being part of the output, uncertainty estimates go on top of the standard semantic predictions. Although this complicates dealing with multiple network outputs, it offers a wider spectrum and deeper insights since the uncertainty could be ignored or considered with various thresholds depending on the situation (Figure 7), for the same trained model and output. Since estimating the uncertainty aims at smoothly quantifying the domain gap from the training data, we believe it is better suited to highly unpredictable unseen real-world scenarios as in holistic segmentation settings.

Furthermore, Figure 8 shows the capability of each method to identify instances of unknown OOD objects. For both approaches, this is related to the clustering of embeddings corresponding to those pixels predicted as unknown, via *void* (OSIS), or as highly uncertain (ours). In particular, OSIS tended to over-fragment unknown objects into several small instances, as seen in the fourth, sixth, eighth, and last images. This again proves our modifications' effectiveness when dealing with the embeddings, as described in Section 4 and evaluated in Table 3. Additionally, OSIS could not distinguish the two neighboring OOD objects in the sixth image. Moreover, OSIS often improperly assigned large regions to the same unknown instance. Similarly to ours, OSIS considers every unknown segment as part of an instance. By learning and predicting the *void* class, during training, OSIS learned to precisely segment the bonnet of the ego car (labeled as *void* in Cityscapes [7]). However, at test time on Lost&Found it could not tell the ego vehicle bonnet apart from a wide variety of pixels. This was the case for the unknown object in the seventh image, which was entirely assigned to the same instance as the bonnet or many other segments around knowns and unknowns (colored in black). The ego vehicle bonnet unknown instance (black) often surrounded other predicted unknown instances (e.g., in the second, fourth, sixth, eighth, and last images).

A benefit of estimating the uncertainty is the ability to account for a wide array of unusual regions. This is valuable for downstream tasks, e.g., trajectory prediction and path planning. Specifically, uncertainty estimates by the proposed U3HS were high on the stroller in Figure 1, as well as in Figure 4 on the walking assistance device on the left of the upper image and the cart pushed by the man waving on the right of the bottom image in Figure 9, none of which were labeled as unknown in the dataset [18], as they were not part of the objects manually placed by the authors. In Figure 8, this is repeated from a different perspective on the stroller in the background of the second image, the unusual van with the open doors in the fourth image, and the duffel bag in the sixth. By learning and predicting *void*, OSIS ignored these unusual regions as it lacks the flexibility and granularity that our U3HS offers by estimating the uncertainty.

A.5.2 Additional Results on Unknowns

Lost&Found Figure 9 shows additional qualitative outputs. Once again, it can be seen how challenging the proposed holistic segmentation setting is. As in the predictions of Figure 4, the model can distinguish most unknown objects. It can be seen how specific areas of the images trigger higher uncertainty estimates. This is the case of the fences in the second and third images of Figure 9, as well as unknown objects not part of the OOD labels of Lost&Found, such as the cart on the right of the bottom image, as previously mentioned. As previously seen, *stuff* structures (e.g., fences) are assigned to a single coherent instance ID throughout the whole image. At the same time, unusual objects (e.g., the cart in the last picture) have their dedicated ID. Figure 9 also provides some examples of unusual scenes present in the Lost&Found [18] dataset, posing significant challenges compared to Cityscapes [7].

MS COCO Figures 10 and 11 show qualitative outputs of the proposed U3HS on the held out classes of MS COCO [16]. The images report the vast diversity of the dataset, ranging from outdoor scenes to indoor close-ups. Remarkably, U3HS delivered precise segments for unknown objects, correctly segmenting instances of individual unknowns despite their similarity with other objects of the same type in the same input, with reasonable estimations of known classes too (Figure 10). Due to the difficulty of this problem, only a handful of segments are perfect, leaving room for improvements for known and unknown objects. Thanks to its strong uncertainty estimation capabilities, the proposed U3HS not only identified the held-out classes but also other unknown objects which are not part of the set of known classes, such as the rice cooker and the umbrella on the right of Figure 11 in the third and bottom rows respectively. This shows the efficacy of our method on a wide variety of scenarios typical of the real world.

A.5.3 Failure Cases

MS COCO While the proposed U3HS delivers reasonable estimates in various settings, from indoor to outdoor, the problem at hand is highly challenging, and its predictions are not perfect, as confirmed by the quantitative results. Figure 12 reports failure cases on a set of challenging samples

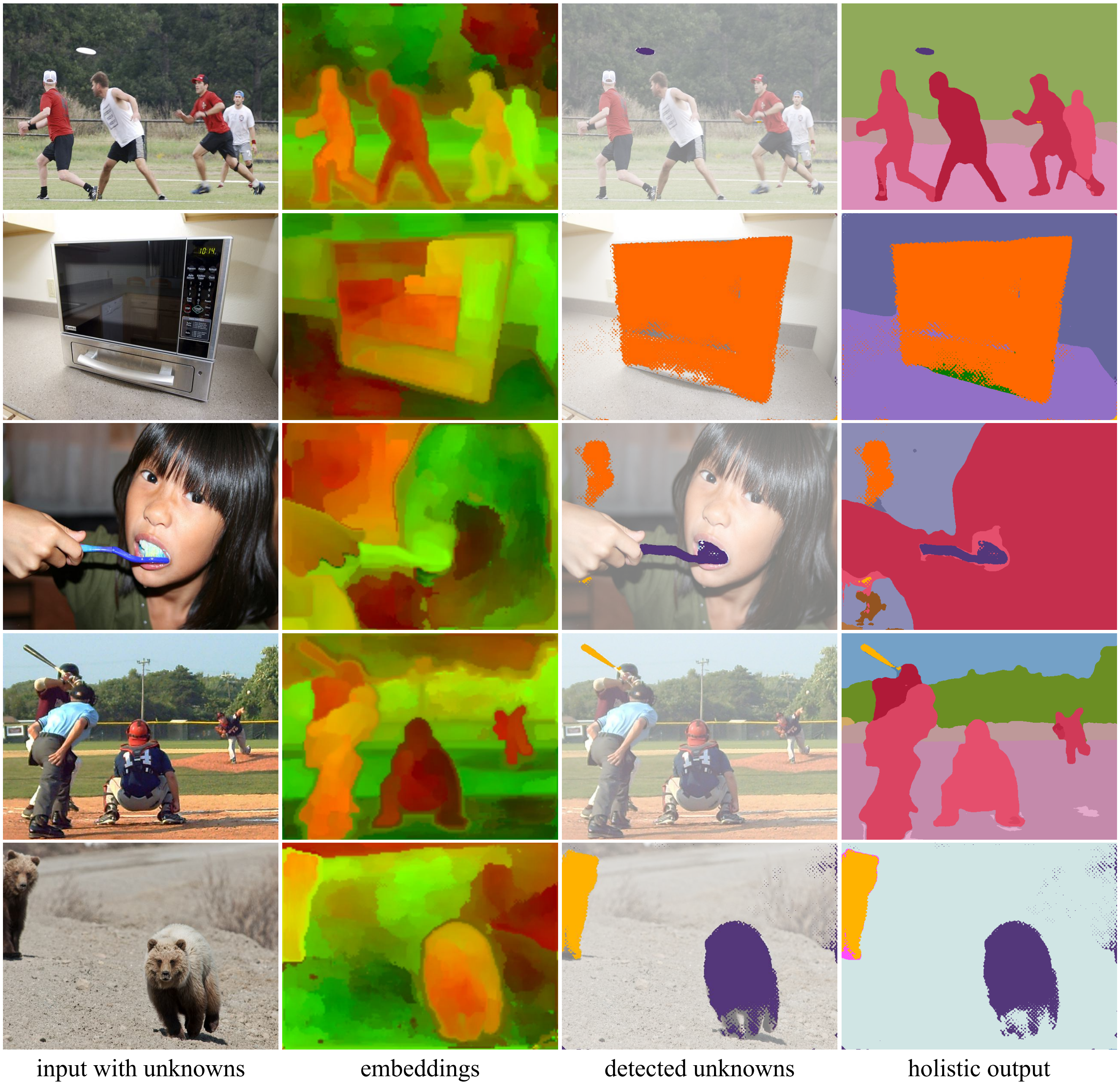


Figure 10. Example predictions of the proposed U3HS on OOD data containing the held out classes of MS COCO [16]. Held out classes (unseen unknowns) in the samples: *frisbee*, *microwave*, *toothbrush*, *baseball bat*, and *bear*. All samples are equally resized. Input, embeddings, detected unknowns, and holistic output are shown.

of MS COCO. It can be seen that while U3HS identified unknowns reasonably, it often missed those objects that are part of the evaluated held-out categories. A series of issues cause this. In some instances, the integrated uncertainty estimates could not fully discover the unseen unknowns, e.g., only partially detected for both *refrigerators* at the left of the third and fourth rows. We also noticed systematic issues with certain classes, such as *keyboard*, *mouse* (both in the right of the fourth row), *hot dog* (in the fifth row), and *scissors* (right of the last row). This could be attributed to these objects being semantically relatively close to known classes, such as *sandwich* for *hot dog*. Small objects were hard to see and therefore ignored by our method. This is the case of the *toothbrush* behind the *cat* in the last row of the figure. U3HS also had difficulties telling apart from one another very close and similar unknowns, e.g., the central *frisbees* in the right of the second row differing for the logo's color. However, it could successfully separate neighboring

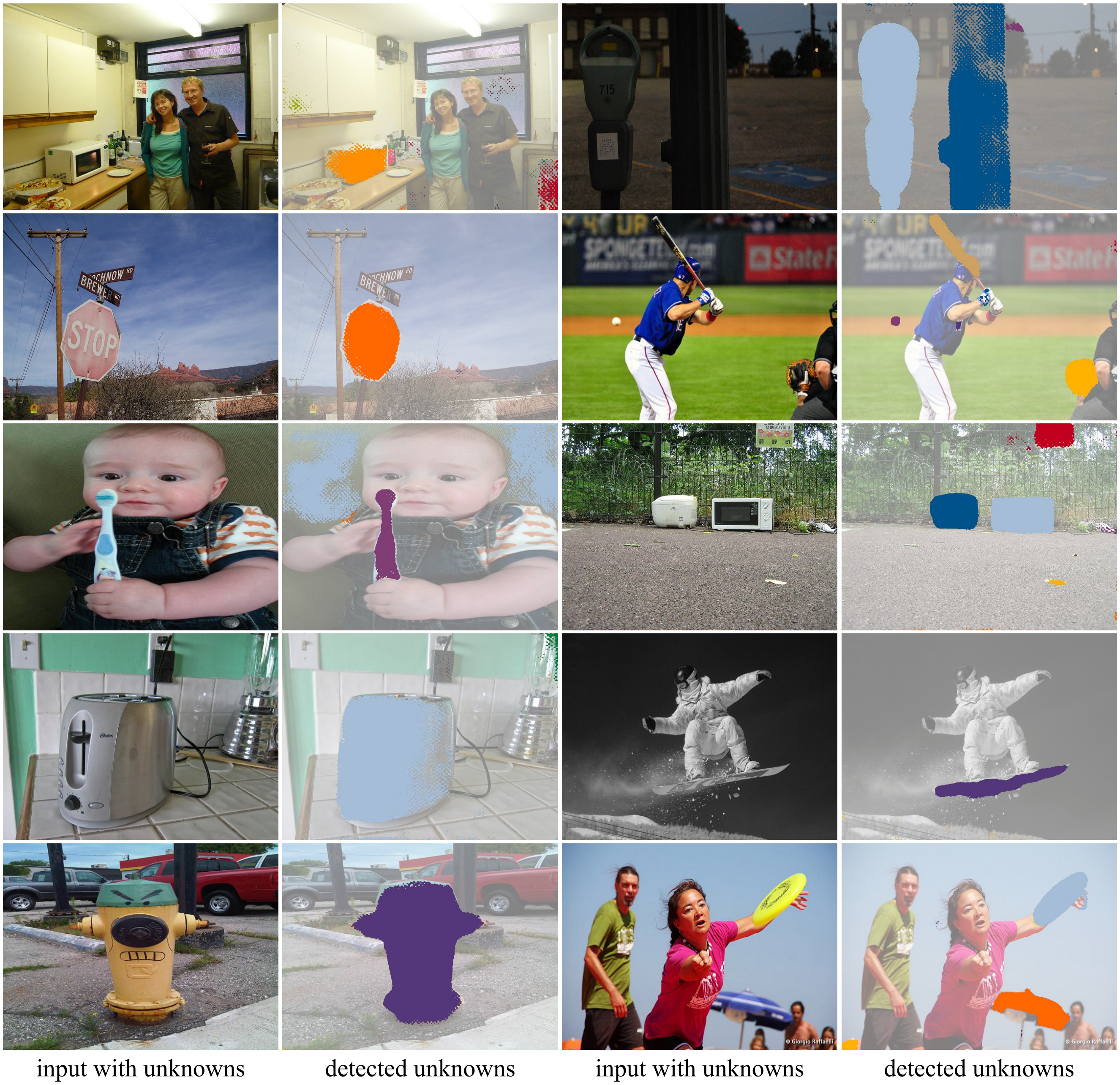


Figure 11. Example predictions of the proposed U3HS on OOD data containing the held out classes of MS COCO [16]. Held out classes (unseen unknowns) in the samples: *microwave*, *parking meter*, *stop sign*, *baseball bat*, *toothbrush*, *toaster*, *snowboard*, *fire hydrant*, and *frisbee*. Other unknown objects are included in the samples, such as the umbrella and the rice cooker (i.e., not part of the known classes). All samples are equally resized. Input and detected unknowns are shown.

objects on multiple occasions, such as the containers on the ground of the left image in the fourth row (not evaluated in this experiment). Moreover, we noticed difficulties with particularly unusual inputs and cluttered environments, e.g., on the left of the second row. With other inputs, the uncertainty of U3HS was also triggered on known objects, such as the *dogs* on the top right (albeit correctly separated into two instances), or the cabinets on the right of the third row. Since several held-out classes contained everyday kitchen-related items (e.g., *refrigerator*, *microwave*, and *toaster*), or typical desk objects (e.g., *keyboard* and *mouse*), the model could see only a handful of kitchens and offices (i.e., those images where none of these held out objects appeared), which severely impacted its ability to handle these situations appropriately. This is a limitation of holding out samples from MS COCO, compared to using a dedicated sepa-

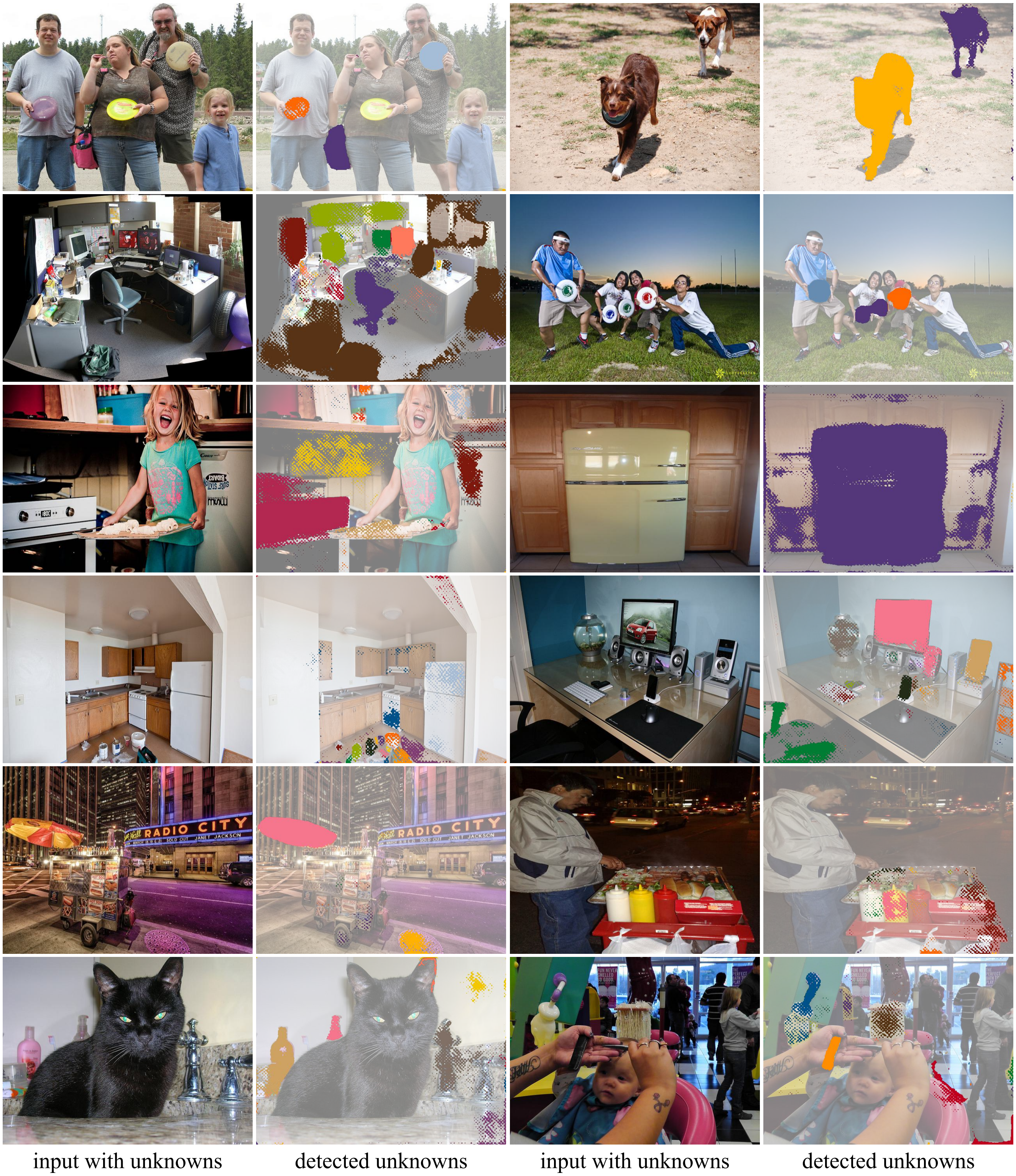


Figure 12. Failure predictions of the proposed U3HS on OOD data containing the held out classes of MS COCO [16]. Held out classes (unseen unknowns) in the samples: *frisbee*, *keyboard*, *mouse*, *refrigerator*, *hot dog*, *toothbrush*, and *scissors*. Other unknown objects are included in the samples, such as bag and comb (i.e., not part of the known classes). All samples are equally resized. Input and detected unknowns are shown.

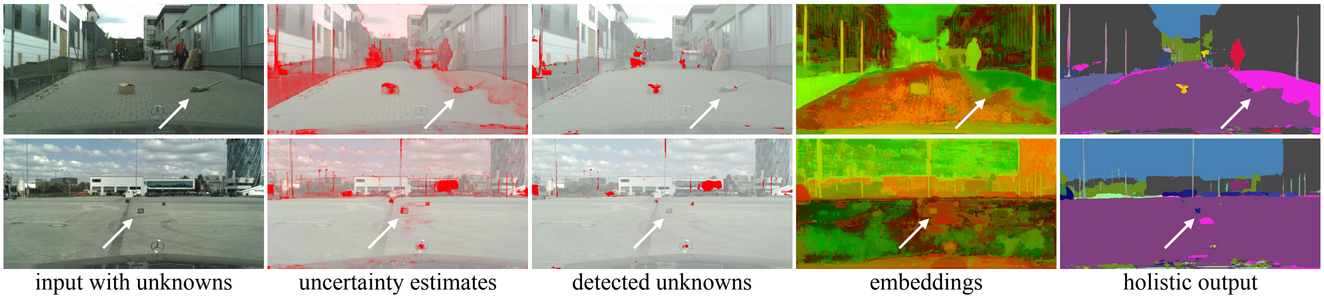


Figure 13. Failure predictions of the proposed U3HS on OOD data from the test set of Lost&Found [18], with a transfer from Cityscapes [7]. White arrows mark missed OOD objects as the estimated uncertainty was relatively low and filtered out.

rate dataset, such as Lost&Found [18]. Furthermore, given that U3HS estimates the model uncertainty, more training data covering a wider variety of scenarios could be beneficial to further reduce the uncertainty on the known classes and improve the known-unknown boundary of U3HS.

Lost&Found Figure 13 shows failure cases caused by the necessary filtering of the uncertainty estimates. While the uncertainty was triggered by a variety of unusual areas, including the vast majority of unknown objects, its a priori filtering (based on closed-set training data, Section 4.2) sometimes caused the unknown object to be completely undetected. Although this filtering is aimed at removing low uncertainty areas which are probably in-domain (e.g., the fence in the upper image), it could inadvertently remove proper OOD objects (e.g., those marked by the white arrows). This is related to the trade-off shown in Figure 7, so keeping more unknowns (i.e., lower threshold t) reduces the in-domain performance. Nevertheless, in the embeddings visualizations, the model correctly isolated the entire marked box in the lower image and precisely segmented the cardboard box in the upper one. However, the two unknown objects were not detected, due to the difficulty of merging multiple outputs and interpreting uncertainty estimates without access to OOD data. It should be considered that the proposed U3HS does not distinguish between the uncertainty for unknown objects and that of unusual known classes. The difference might lie in the amount of uncertainty corresponding to these regions, hence the filtering via the threshold t to attempt telling apart completely unknown from unusual, which remains highly challenging without using any information about unknowns at training time, as in our setup.

References

[1] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes benchmark: Measuring blind spots in semantic segmentation. *Springer International Journal of Computer Vision*, 129(11):3119–3135, 2021. 2

[2] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15333–15342, 2021. 4, 5

[3] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMeIfYouCan: A benchmark for anomaly segmentation. In *Neural Information Processing Systems - Datasets and Benchmarks Track*, 2021. 1, 2

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818. Springer, 2018. 4

[5] 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. 1

[6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1, 2, 3, 4, 5, 6, 7, 8, 12

[8] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 3

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996. 3

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. 4, 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016. 4, 5

[12] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7031, 2022. 2

[13] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1184, 2021. 2, 4

[14] Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized Max Logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15425–15434, 2021. 1, 4, 5

[15] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3D-VField: Adversarial augmentation of point clouds for domain generalization in 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17295–17304, 2022. 1, 2, 8

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 3, 4, 8, 9, 10, 11

[17] Jeremiah Z. Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, 2020. 3, 4, 5

[18] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting small road hazards for self-driving vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1099–1106, 2016. 2, 3, 4, 5, 6, 7, 8, 12

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3

[20] Murat Sensoy, Lance M. Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing*, pages 3183–3193, 2018. 4, 5

[21] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020. 4, 5

[22] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *Proceedings of the Conference on Robot Learning*, pages 384–393. PMLR, 2020. 2, 3, 4, 5, 6, 7

[23] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *Proceedings of the European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2

[24] Le You, Han Jiang, Jinyong Hu, C Hwa Chang, Lingxi Chen, Xintong Cui, and Mengyang Zhao. GPU-accelerated faster Mean Shift with Euclidean distance metrics. In *Proceedings of the Computers, Software, and Applications Conference*, pages 211–216. IEEE, 2022. 6