

# Unifying Training and Inference for Panoptic Segmentation

Qizhu Li Xiaojuan Qi\* Philip H.S. Torr  
University of Oxford

{qizhu.li, xiaojuan.qi, philip.torr}@eng.ox.ac.uk

## Abstract

We present an end-to-end network to bridge the gap between training and inference pipeline for panoptic segmentation, a task that seeks to partition an image into semantic regions for “stuff” and object instances for “things”. In contrast to recent works, our network exploits a parametrised, yet lightweight panoptic segmentation submodule, powered by an end-to-end learnt dense instance affinity, to capture the probability that any pair of pixels belong to the same instance. This panoptic submodule gives rise to a novel propagation mechanism for panoptic logits and enables the network to output a coherent panoptic segmentation map for both “stuff” and “thing” classes, without any post-processing. Reaping the benefits of end-to-end training, our full system sets new records on the popular street scene dataset, Cityscapes, achieving 61.4 PQ with a ResNet-50 backbone using only the *fine* annotations. On the challenging COCO dataset, our ResNet-50-based network also delivers state-of-the-art accuracy of 43.4 PQ. Moreover, our network flexibly works with and without object mask cues, performing competitively under both settings, which is of interest for applications with computation budgets.

## 1. Introduction

As a pixel-wise classification task, panoptic segmentation aims to achieve a seamless semantic understanding of all countable and uncountable objects in a scene - *a.k.a.* “things” and “stuff” respectively, and delineate the instance boundaries of objects where semantically possible.

While early attempts at tackling panoptic segmentation often resort to two separate networks for instance and semantic segmentation, recent works [18, 16, 13, 26, 27] are able to improve the overall efficiency by constructing the two branches on a single, shared feature extractor, and training the multi-head, multi-task network jointly. However, these works have stopped short of devising an end-to-end pipeline for panoptic segmentation, as they all adopt a post-

processing stage with heuristics to combine the different outputs of their multi-task networks, following [14, 13]. Such pipelines suffer from several shortcomings. Firstly, post-processing often requires a time-consuming trial-and-error procedure to mine a good set of hyperparameters, which may need to be repeated for each image domain. As the performance of an algorithm can be quite sensitive to the choice of hyperparameters, how well a method performs can quickly degenerate to a function of the amount of computation resources at its disposal [15, 13]. Secondly, methods without an explicit loss function for panoptic segmentation [18, 16, 13, 27] cannot directly optimise for the ultimate goal. Even with expert knowledge, it is difficult to design an exhaustive set of rules and remedies for all failure modes. An example is shown in Fig. 1 (c): after the heuristic post-processing, the missing part of the car cannot be recovered.

To achieve an end-to-end system, we reckon three challenging steps need to be taken: (1) unify the training and inference, enabling the network to *differentiably* produce panoptic segmentation during training; (2) embed a data-driven mechanism in the multi-task network whereby imperfect and coarse cues can be cleaned and corrected; (3) design an appropriate loss function to directly optimise the global objective for panoptic segmentation.

To achieve (1) and (2), we propose a novel pipeline using segmentation and localisation cues to predict a coherent panoptic segmentation in an end-to-end manner. At the heart of this pipeline lie a *dynamic potential head* – a parameter-free stage that represents a dynamic number of panoptic instances, and a *dense instance affinity head* – a parametrised, efficient, and data-driven module that predicts and utilises the likelihood for any pair of pixels to belong to the same “thing” instance or “stuff” class. These two differentiable heads produces full panoptic segmentation during training and inference, eradicating the train-test logic discrepancy.

Furthermore, to fulfil (3), we propose a *panoptic matching loss* which computes loss directly on panoptic segmentations. This objective function, together with the differentiable nature of our proposed panoptic head, enables the network to learn in an end-to-end manner. To our best knowledge, our loss is the first to perform online segment matching before

\*Xiaojuan Qi is now with the University of Hong Kong.

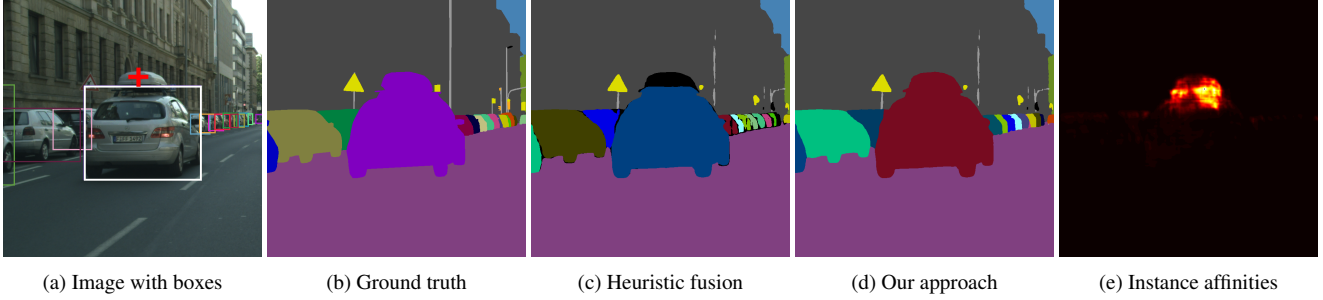


Figure 1. Comparison of our approach vs. the heuristic rule-based method of [13]. We overlay the predicted bounding boxes on the input images for visualisation. For the cross-marked pixel in (a) which falls outside its bounding box, we show its instance affinities in (e). Heuristics-based fusion [13] produces truncated objects when localisation is not accurate, while our instance affinity enables the network to recover the full object, by propagating information between pixels with strong instance affinities. Best viewed in colour.

computing a cross entropy loss in an end-to-end panoptic segmentation system. The matching step allows training the network with *predicted* detections, thereby incentivising it to handle imperfect localisation cues. While the idea is not convoluted, our ablation studies (Table C, Supplementary) show that doing so – as opposed to training with ground truth detections – yields performance gains.

By closing the gap between training and inference, the network enjoys improved accuracy in challenging scenarios. As illustrated in Fig. 1, by aggregating panoptic logits across the whole image according to the predicted affinity strengths (Fig. 1e), our parametrised panoptic head is able to fix inaccurate predictions from a previous stage - truncated objects due to imperfect bounding box localisations (Fig. 1c).

Last but not least, thanks to its power of improving coarse panoptic logits, our network achieves competitive performance even without using object mask cues, which are required in most recent approaches [18, 16, 13, 26]. This means our method can offer an additional degree of flexibility in terms of network design, a trait desirable for applications with a limited computation and time budget. On the challenging Cityscapes and COCO datasets, our models set new records for ResNet-50-based networks, achieving panoptic qualities (PQ) of 61.4 and 43.4 respectively.

## 2. Related work

Arguably, the problem of panoptic segmentation can be viewed as a combination of instance and semantic segmentation. Indeed, this interpretation has guided many recent works on panoptic segmentation [14, 13], where it is largely approached as a bi-task problem, and the focus is placed on solving both sub-problems simultaneously and efficiently. Shared features of these works include the use of networks with multiple specialised subnets for each sub-task, and the lack of an explicit objective on panoptic segmentation.

In addition to the inclusion of “stuff” classes, another major difference between panoptic and instance segmentation is that the former requires all pixels to be given a unique label,

whereas the latter does not. As a result, “thing” predictions from an off-the-shelf detection-driven instance segmentation network – *e.g.*, Mask-RCNN [9] – cannot be readily inserted into the panoptic prediction, as pixels need to have their conflicting instance labels resolved. Moreover, contradictions between the semantic and instance branch must also be carefully resolved. This prompted recent works to adopt an offline postprocessing step first described in [14] to perform conflict resolution and merger of instance and semantic predictions, based on a set of carefully tuned heuristics. A number of works have also attempted to encourage consistency between semantic and instance predictions by adding a communication mechanism between the two subnets [16, 18]. However, as these proposed changes do not modify the output format of the network, they still rely on postprocessing to produce panoptic predictions. In addition, Liu *et al.* proposes to directly learn the ordering of “thing” instances for conflict resolution [20]. However, this approach does not handle overlapping instances pixel-by-pixel – as it predicts a single ranking score for each instance – and does not reconcile conflicts between “stuff” and “thing”.

A small number of works have attempted to advance towards an end-to-end network with a unified train-test logic. We observe that [17] extends a dynamically instantiated instance segmentation network described in [1] to solve the panoptic segmentation problem. It produces non-overlapping segments by design, and is trained end-to-end, given detections. However, it is prone to failures when objects of the same class are nearby and similarly coloured. Moreover, its Instance CRF suffers from the very small number of trainable parameters (since the compatibility transforms are frozen as the Potts model), and is made less attractive by the need to grid search good kernel variances for the bilateral filters in the message passing step.

Recently, Xiong *et al.* [26] modifies the unary terms of [1, 17] and proposes a parameter-free, differentiable panoptic head to fuse semantic and instance segmentation predictions during training. Similar to [17], it allows a panoptic loss

to be directly applied on the fused probabilities. However, in the inference phase, it still resorts to several heuristic strategies (e.g., overlap-based instance mask pruning) and relies on a complex voting mechanism to determine the semantic categories of predicted segments, deviating from a unified training and inference pipeline. Furthermore, the effectiveness of their parameter-free panoptic head heavily depends on the quality of semantic and instance predictions it receives, since it arguably functions as an online heuristic merger due to the absence of learnable weights.

Also pertinent to this work is the extensive research carried out around the techniques of long-range contextual aggregation. Aside from CRF-driven methods [15, 29, 1], Bertasius *et al.* proposes a semantic segmentation method based on random walks to learn and predict inter-pixel affinity graphs, and iteratively multiply the learnt affinity with an initial segmentation to achieve convergence [3]. Lately, another technique, self-attention, has been successful in several vision tasks [24, 28, 7]. However, its quadratic memory and computation complexity has cast doubt over its practicality. To mitigate this problem, Shen *et al.* [23] suggests to invoke the associativity of matrix multiplication and avoid the explicit production of expensive attention maps. This approach effectively reduces the complexity to a linear one,  $O(HW)$ , making it suitable for pixel-level labelling tasks.

Albeit sharing certain operational similarities with self-attention and non-local methods [28, 12, 24], our proposed dense instance affinity head serves a different purpose, and cannot be substituted by directly inserting these operations in the backbone. The aforementioned methods work by enhancing the expressiveness of extracted features, as reflected in the fact that these actions are performed in the feature space, and can generally lead to performance gains for many tasks. In contrast, our proposed instance affinity is not a generic feature enhancer. It is specifically designed and tasked to model the pairwise probability for any two pixels to belong in the same “thing” instance or “stuff” category. This relationship in turn enables our network to revise and resolve. With this purpose in mind, we incorporate insights from [23] to construct a module that is lightweight, learnable, and agnostic to the number of channels, allowing us to model a dynamic number of instances across different images.

### 3. Proposed approach

Our proposed network (Fig. 2) consists of four blocks. A shared *fully convolutional backbone* extracts a set of features. Operating on these features, a *semantic segmentation submodule* and an *object detection submodule* produce segmentation and localisation cues, which are fused and revised by the proposed *panoptic segmentation submodule*. All components are differentiable and trained jointly, end-to-end.

#### 3.1. Backbone

The pipeline starts with a shared fully convolutional backbone, which takes an input image of spatial dimension  $H \times W$ , and generates a set of features  $F$ . In our experiments, we adopt a simple ResNet-FPN backbone that outputs four multi-scale feature maps [19], following a common practice in prior works [13, 26]. To encourage global consistency, we carry out a squeeze-and-excitation operation [11] on the top-level ResNet feature before producing the first FPN feature. A similar strategy is used in [26].

#### 3.2. Semantic segmentation submodule

The backbone features  $F$  are fed into the semantic segmentation submodule to produce a  $\frac{H}{d} \times \frac{W}{d} \times (N_{st} + N_{th})$  tensor  $V$ , where  $N_{st}$  and  $N_{th}$  are the number of “stuff” and “thing” classes respectively.  $V_i(l)$  denotes the probability that pixel  $p_i$  belongs to semantic class  $l$ . The spatial dimension is downsampled  $d$  times to strike a balance between resolution and complexity. We choose  $d$  as 4 in the experiments.

Multiple implementations for this submodule have been proposed in the literature, all showing decent performance [13, 26]. In this work, we modify the design in [26] by inserting a Group Normalisation operation [25] after each convolution, which has been observed to help stabilise training. Please refer to the supplementary for further details.

#### 3.3. Object detection submodule

In parallel, the features  $F$  are also passed to an object detection submodule, which generates  $D$  object detections, consisting of bounding boxes  $B = \{B_1, B_2, B_3, \dots, B_D\}$ , confidence scores  $s = \{s_1, s_2, s_3, \dots, s_D\}$ , and predicted classes  $c = \{c_1, c_2, c_3, \dots, c_D\}$ . Additionally, we add a whole image bounding box for each “stuff” class to the object detection predictions, raising the total number of detections to  $D + N_{st}$ . Doing so allows the panoptic submodule to process “things” and “stuff” with a unified architecture.

Notably, the versatility of the panoptic submodule allows our network to work with or without object masks. When the object detection submodule has the capability to predict instance masks for “things”  $M = \{M_1, M_2, M_3, \dots, M_D\}$ , they are easily incorporated into the dynamic potential  $\Psi$ . Details will be given in Sec. 3.4.1.

#### 3.4. Panoptic segmentation submodule

This submodule serves as the mastermind of the pipeline. Receiving cues from the two prior submodules, the panoptic segmentation submodule combines them into a dynamic potential  $\Psi$  (Sec. 3.4.1) and revises it according to predicted pairwise instance affinities (Sec. 3.4.2), producing the final panoptic segmentations with the same logic in training and inference. This pipeline is illustrated in Fig. 3.

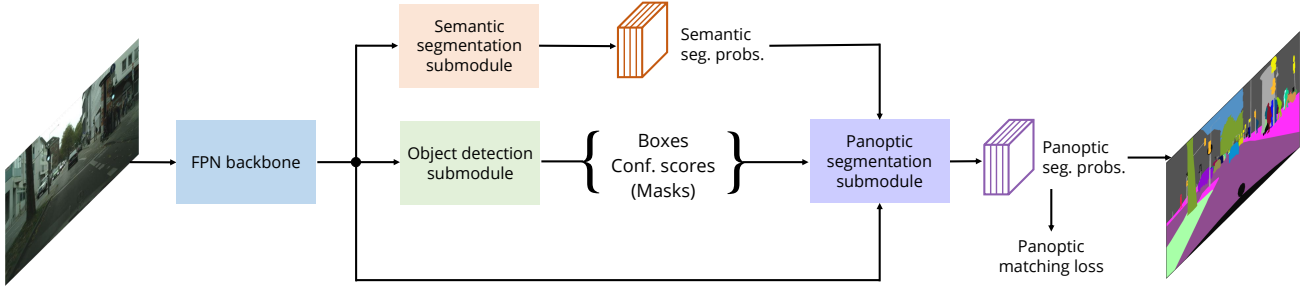


Figure 2. Overview of the network architecture. Semantic segmentation and object detections are fed into the proposed panoptic segmentation submodule – including a dynamic potential head and a dense instance affinity head – to produce panoptic segmentation predictions without requiring post-processing. All components are differentiable, and the network is trained end-to-end.

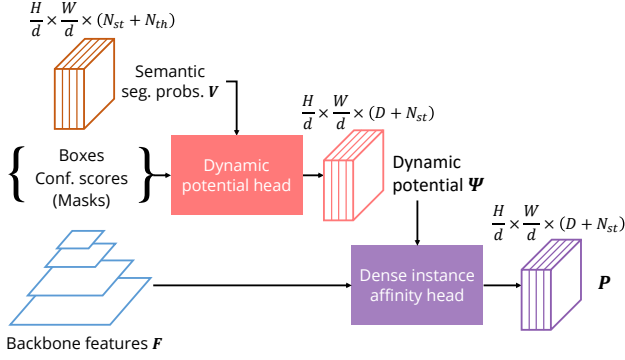


Figure 3. The panoptic segmentation submodule. Details on the dynamic potential head and dense instance affinity head are further clarified in Fig. 4 and 5 respectively.

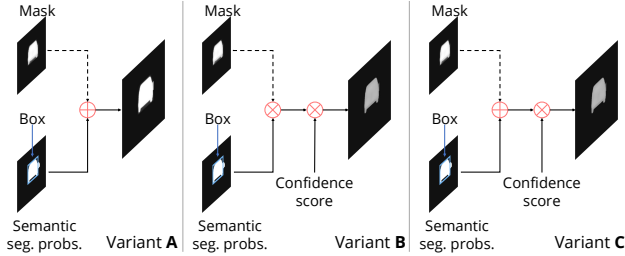


Figure 4. Three variants of the dynamic potential head. For clarity, we only show one instance in each diagram. In practice, the same operation is extended to all detections and “stuff”. Note that the dotted path is only activated when masks are provided to the head. When no masks are given, variant B and C are equivalent.

### 3.4.1 Dynamic potential head

The dynamic potential head functions as an assembly node for segmentation and localisation cues from prior submodules. This head is capable of representing varying numbers of instances as it outputs a *dynamic* number of channels, one for each object instance or “stuff” class. We present three variants of dynamic head design, as illustrated in Fig. 4. Variant A is proposed in [26], whereas the mask-free parent of B and C is first described in [1] as the box consistency term. A main difference between variant A and the rest is the

absence of detection score in A. We argue that leveraging detection scores can suppress false positives in the final output, as unconfident detections will be attenuated by its score. Thus, we will describe variant B and C in more details.

Given  $(D + N_{st})$  bounding boxes  $B$  and box classes  $c$  (including the dummy full-image “stuff” boxes), it populates each box region with a combination of semantic segmentation probabilities  $V$  and box confidence scores  $s$  to produce a *dynamic potential*  $\Psi$  with  $(D + N_{st})$  channels:

$$\Psi_i(k) = \begin{cases} s_k V_i(c_k) & \text{for } i \in B_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Optionally, if provided with object masks  $M$ , the dynamic potential head can also incorporate them into  $\Psi$ . Defining  $M$  to be image-resolution instance masks where the raw masks have been resized to their actual dimensions and pasted to appropriate spatial locations in image, the dynamic potential with object masks can be summarised as:

$$\Psi_i(k) = \begin{cases} s_k [V_i(c_k) \odot M_i(k)] & \text{for } i \in B_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In variant B and C, operator  $\odot$  is multiplication and summation respectively. More analysis of the variant B and C are included in the supplementary.

### 3.4.2 Dense instance affinity head

We observe that the dynamic potential  $\Psi$  often carries conflicts and errors due to imperfect cues from semantic segmentation and object localisation. This motivates the design of this parametrised head, with the aim to enable a data-driven mechanism that resolves and revises the output of the dynamic potential head. The main difficulty with injecting parameters into an instance-level head is the varying number of instances across images, which practically translates to a dynamic number of channels in the input tensor. On the other hand, the fundamental building block of a convolutional neural network – convolution – is designed to handle a fixed



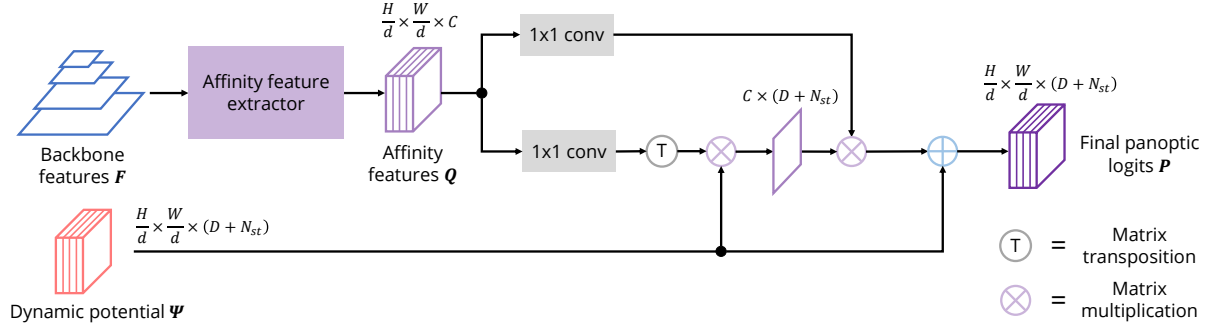


Figure 5. The dense instance affinity head. It is parametrised, expressive, lightweight, and fully differentiable.

number of input channels. This apparent incompatibility has led prior works on panoptic segmentation to use either no parameter at all [26], or only single scaling factors for entire tensors [17] providing limited modelling capacity.

This conundrum can be tackled by driving this head with a pairwise dense instance affinity, which is predicted from data, fully differentiable, and compatible with a dynamic number of input channels. By integrating global information according to the pairwise affinities, it produces the final panoptic segmentation probabilities, from which inference can be trivially made with an  $\text{argmax}$  operation along the channel dimension. Thus, it is amenable to a direct panoptic loss, an ingredient of an end-to-end network.

To construct the dense instance affinity, this head first extracts from the backbone features  $F$  a single feature tensor  $Q$  of dimension  $\frac{H}{d} \times \frac{W}{d} \times C$ , where  $C$  is the number of feature channels, and  $d$  is a downsampling factor. This corresponds to the affinity feature extractor in Fig. 5. The spatial dimensions of  $Q$  can be easily collapsed to produce a  $\frac{HW}{d^2} \times C$  feature matrix.

Normally, the pairwise instance affinities  $A$  – a large  $\frac{HW}{d^2} \times \frac{HW}{d^2}$  matrix – would then be produced by performing a matrix multiplication  $A = QQ^T$ . This would be followed by multiplying  $A$  with a  $\frac{HW}{d^2} \times C'$  input tensor to complete the process. It is, however, prohibitively expensive due to the quadratic complexity with respect to  $HW$ . In a typical training step, where  $(H, W) = (800, 1300)$  and  $d = 4$ , a single precision matrix with the size of  $A$  would occupy 15.7GB of GPU memory, making this approach unpractical.

Drawing from insight of [23], we design a lightweight pipeline for computing and applying the dense instance affinities (Fig. 5). Instead of sequentially computing  $QQ^T\Psi$  which explicitly produces  $A$ , we compute  $Q(Q^T\Psi)$ , since:

$$(QQ^T)\Psi = Q(Q^T\Psi) \quad (3)$$

The result of  $Q^T\Psi$  is a very small  $C \times (D + N_{st})$  tensor, taking only tens of kilobytes. In terms of computation, using the same  $H, W, d$  as the example above and  $(C, D, N_{st} = 128, 100, 53)$  as typically used in experiments, the efficient implementation reduces the total number of multiply-adds by 99.8% to 5 billion FLOPS. For reference, a ResNet-50-FPN

backbone at the same input resolution requires 140 billion FLOPS.

Finally, we add the product back to the input, forming a residual connection to ease the learning task. As such, the full action of our dense instance affinity applier can be summarised with the following expression:

$$P = \Psi + \phi_0(Q)(\phi_1(Q^T)\Psi) \quad (4)$$

where  $\phi_0$  and  $\phi_1$  are each a  $1 \times 1$  convolution followed by an activation. From this formulation, inference is straight forward and does not require any post-processing, as an  $\text{argmax}$  operation on  $P$  along the channel direction readily produces the panoptic segmentation prediction.

Note that we do not compute a loss directly over  $Q$ ; instead, the instance affinities are implicitly trained by supervision from the panoptic matching loss described in the next section. In the preliminary experiments, we tried directly supervising  $Q$  with a contrastive loss, but did not observe performance gains. This shows that our end-to-end training scheme with the panoptic matching loss is already able to guide the model to learn effectively. Detailed discussion of the dense instance affinity operation, with ablation studies and visualisations, is provided in Sec. 4.1.

For simplicity, the affinity feature extractor adopts the same architecture as our semantic segmentation submodule. We use  $C = 128$  in all experiments.

### 3.5. Panoptic matching loss

For instance-level segmentation, different permutations of the indices in the segmentation map are qualitatively equivalent, since the indices merely act to distinguish between each other, and do not carry actual semantic meanings.

During training, we feed predicted object detections into the panoptic segmentation submodule. As a result, the indices of the instances are not fixed or known before hand. To compute loss, we first match the ground truth segmentation to the predicted detections by maximising the intersection over union between their bounding boxes (box IoU). Given a set of  $\alpha$  ground truth segments  $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \dots, \mathcal{T}_\alpha\}$ , and a set of  $\beta$  predicted bounding

boxes  $B = \{B_1, B_2, B_3, \dots, B_\beta\}$ , we find the “matched” ground truth  $\mathcal{T}^*$  which satisfies:

$$\mathcal{T}^* = \underset{\mathcal{Z} \in \pi(\mathcal{T})}{\operatorname{argmax}} \operatorname{IoU}_t(\operatorname{box}(\mathcal{Z}), B) \quad (5)$$

where  $\operatorname{box}(\cdot)$  extracts tight bounding boxes from segments,  $\pi(\mathcal{T})$  refers to all permutations of  $\mathcal{T}$ , and  $t$  sets the minimum match threshold for a match to qualify as valid. Note that the box IoU between different semantic classes are taken to be 0, and  $\alpha$  and  $\beta$  need not be the same. Ground truth segments without matched predictions are set to the “ignore” label, and detections matching to the same ground truth segment are all removed except the top match, before being fed into the panoptic submodule. Both cases do not contribute any gradients. With the “matched” ground truth segmentation  $\mathcal{T}^*$ , we can compute the loss on the predicted panoptic segmentation probabilities  $P$  as per normal with a cross-entropy loss. Our experiments use 0.5 for  $t$ .

Unlike ours, the panoptic loss used by [26] does not have the matching stage and its panoptic head is trained with ground truth detections instead. As a result, the models of [26] are not trained to handle imperfect localisations. In addition, our loss differs from [20] as the loss used by their *spatial ranking module* does not directly supervise panoptic segmentation, does not take “stuff” into account, and thus does not globally optimise in an end-to-end way.

## 4. Experimental evaluation

**Cityscapes.** The Cityscapes dataset features high resolution road scenes with 11 “stuff” and 8 “thing” classes. There are 2,975 training images, 500 validation images, and 1,525 test images. We report on its validation set and test set.

**COCO.** The COCO panoptic dataset has a greater number of images and categories. It features 118k training images, 5k validation images, and 20k *test-dev* images. There are 133 semantic classes, including 53 “stuff” and 80 “thing” categories. We report on its validation set and *test-dev* set.

**Evaluation metric.** Our main evaluation metric is the panoptic quality (PQ), which is the product of segmentation quality (SQ) and recognition quality (RQ) [14]. SQ captures the average segmentation quality of matched segments, whereas RQ measures the ability of an algorithm to correctly detect objects.

We also report the mean Intersection over Union (IoU) score of our initial category-level segmentation  $V$ , and the box Average Precision ( $AP_{box}$ ) of our predicted bounding boxes  $B$ . Additionally, for models which predict object instance masks  $M$  in the object detection submodule, we report its mask Average Precision ( $AP_{mask}$ ) as well. Both  $AP_{box}$  and  $AP_{mask}$  are averaged across IoU thresholds between 0.5 and 0.95, at increments of 0.05.

**Cityscapes training.** We follow most of the learning settings described in [13]. We distribute the 32 crops in a minibatch over 4 GPUs instead. The weights for the detection, semantic segmentation, and panoptic segmentation losses are set to 0.25, 1.0, and 1.0 respectively.

**COCO training.** We follow most of the learning settings for COCO experiments in [13]. For the learning schedule, we train for 200k iterations with a base learning rate of 0.02, and reduce it by a factor of 10 at 150k and 190k iterations. While this learning schedule differs from that used in [13], we found that our panoptic submodule with its additional parameters benefits from the new schedule. In terms of loss weights, we use 1.0, 0.2, and 0.1 for the object detection, semantic segmentation, and panoptic segmentation losses.

### 4.1. Ablation studies

We conduct detailed ablation studies for five different settings, including two architecture choices (msk. and aff.), one training strategy (e2e.), and two inference options (heu. and amx.). We report the results in Table 1. Explanations for the abbreviations can be found in the table caption. For clarity, we provide a brief description of the ablation models:

- Model  $\mathbb{A}$  uses a Faster-RCNN head as its object detection submodule, and has neither the dense instance affinity head nor the panoptic matching loss. The dynamic potential  $\Psi$  is used as the final output  $P$ .
- Model  $\mathbb{B}$  differs from  $\mathbb{A}$  by employing the dense instance affinity head and the panoptic matching loss.
- In  $\mathbb{C}1$  and  $\mathbb{C}2$ , the model uses a Mask-RCNN head as its object detection submodule, and has neither the dense instance affinity head nor the panoptic matching loss. During inference,  $\mathbb{C}1$  merges the semantic and instance segmentation predictions using heuristics [14], whereas  $\mathbb{C}2$  outputs the dynamic potential  $\Psi$  as  $P$ .
- The pair ( $\mathbb{D}1$ ,  $\mathbb{D}2$ ) differs from ( $\mathbb{C}1$ ,  $\mathbb{C}2$ ) by employing the instance affinity and the panoptic matching loss.

Note that model  $\mathbb{A}$  and  $\mathbb{B}$  do not produce nor use object mask predictions, and are therefore not possible to test with the heuristic merger strategy [13]. In addition, the pair  $\mathbb{C}1$  and  $\mathbb{C}2$ , as well as  $\mathbb{D}1$  and  $\mathbb{D}2$ , are identical models using different inference methods.

**Dense instance affinity.** Comparing across model  $\mathbb{A}$  and  $\mathbb{B}$ , it is evident that training and testing with the proposed dense instance affinity leads to significant performance boosts. Increased performances are seen across all metrics, with the largest rises in PQ (+4.4 for all, +4.2 for “things” and +4.4 for “stuff”) and RQ (+4.0). This testifies to the effectiveness of the dense instance affinity, even with only

Model	Settings					PQ			SQ	RQ	IoU	$AP$	$AP$
	msk.	aff.	e2e.	heu.	amx.	all	th.	st.	all	all	all	mask	box
A					✓	54.6	46.0	60.9	77.9	68.4	75.0	–	36.9
B		✓	✓		✓	59.0	50.2	65.3	80.1	72.4	77.8	–	38.1
C1	✓			✓		59.3	51.4	65.0	79.8	73.2	78.1	33.8	38.1
C2	✓				✓	59.6	52.4	64.8	80.4	72.9	78.1	33.8	38.1
D1	✓	✓	✓	✓		60.6	52.4	66.5	80.4	74.2	79.5	33.7	38.8
D2	✓	✓	✓		✓	61.4	54.7	66.3	81.1	74.7	79.5	33.7	38.8

Table 1. Ablation studies on Cityscapes validation set. Settings include two architecture variations: whether to utilise object masks (msk.), and whether to utilise the proposed instance affinity (aff.); one training option: whether to train end-to-end with the panoptic matching loss (e2e.); and two inference strategies: whether to directly take  $\arg\max$  (amx.) of the panoptic logits (which is either  $\Psi$  for A and C2, or  $P$  for B and D2) or use the heuristic merging strategy [13] (heu.).

box predictions. A similar trend is also evident with object masks enabled, between model C2 with D2, recording a 1.8 rise in overall PQ. Fig. 6 visualises some examples of instance affinities, with more in the supplementary materials.

**End-to-end training with panoptic matching loss.** While C1 and D1 are trained differently – with the former being trained jointly, and the latter being trained end-to-end with the panoptic matching loss – they are tested using the same heuristic strategy [13]. Therefore, the 1.3 increase in PQ of D1 over C1 solely stems from the fact that D1 undergoes end-to-end training, and shows that our end-to-end training strategy with the panoptic matching loss is effective.

**Unified training and inference pipeline.** For D1, we test a model trained end-to-end with the panoptic matching loss using the heuristic merger strategies. In contrast, for D2, we take the same model and take  $\arg\max$  from the final panoptic logits. We can see that the D2 still outperforms D1 by 0.8 PQ, giving proof for the benefit of having a unified training and testing pipeline.

## 4.2. Comparison with state-of-the-art

**Cityscapes.** We compare our results with other methods on Cityscapes validation set in Table 2. All entries are ResNet-50 [10] based except [17, 27]. We sort prior works into two tracks, depending on whether the network performs instance segmentation internally. For both tracks, our method achieves the state-of-art. The most telling comparison is between our model and UPSNet, as these methods have a similar network architecture other than our proposed panoptic segmentation submodule. Our network is able to outperform UPSNet by 2.1 PQ. On the other hand, among methods that do not rely on instance segmentation [17, 27], our system outperforms the previous state-of-art by 3.5 PQ, even though they utilise stronger backbones (Xception-71 [5] and ResNet-101 [10]) than ours (ResNet-50).

Speed-wise, our design compares favourably with other state-of-the-art models. On Cityscapes, inference takes

Method	PQ			SQ	RQ	IoU	$AP$	$AP$
	all	th.	st.	all	all	all	mask	bbox
Li <i>et al.</i> [17]	53.8	42.5	62.1	–	–	<b>79.8</b>	–	–
DeeperLab [27]	56.5	–	–	–	–	–	–	–
SSAP [8]	58.4	<b>50.6</b>	–	–	–	–	–	–
Ours (w/o mask)	<b>59.0</b>	50.2	<b>65.3</b>	<b>80.1</b>	<b>72.4</b>	77.8	–	<b>38.1</b>
TASCNet [16]†	55.9	50.5	59.8	–	–	–	–	–
Attention [18]†	56.4	52.7	59.0	–	–	73.6	33.6	–
Pan. FPN [13]†	57.7	51.6	62.2	–	–	75.0	32.0	–
UPSNet [26]†	59.3	54.6	62.7	79.7	73.0	75.2	33.3	<b>39.1</b>
Pan. Deeplab [4]†	59.7	–	–	–	–	–	–	–
Seamless [21]†	60.3	<b>56.1</b>	63.3	–	–	77.5	33.6	–
Ours (w/ mask)†	<b>61.4</b>	54.7	<b>66.3</b>	<b>81.1</b>	<b>74.7</b>	<b>79.5</b>	<b>33.7</b>	38.8

Table 2. Panoptic segmentation results on Cityscapes *val.* set. Models that run instance segmentation internally are marked with †. Other than [17, 27], all works are ResNet-50 [10] based. For fairness, we only include numbers obtained via *single-scale* inference.

Method	PQ			SQ	RQ	IoU	$AP$	$AP$
	all	th.	st.	all	all	all	mask	bbox
JSIS-Net [6]	26.9	29.3	23.3	72.4	35.7	–	–	–
Pan. Deeplab [4]	35.1	–	–	–	–	–	–	–
Pan. FPN [13]	39.0	45.9	28.7	–	–	–	33.3	–
UPSNet [26]	42.5	<b>48.6</b>	33.4	78.0	52.5	<b>54.3</b>	34.3	37.8
Ours (w/ mask)	<b>43.4</b>	<b>48.6</b>	<b>35.5</b>	<b>79.6</b>	<b>53.0</b>	53.7	<b>36.4</b>	<b>40.5</b>

Table 3. Panoptic segmentation results on COCO 2017 validation set. All methods are based on a ResNet-50 backbone.

386ms<sup>1</sup> and 201ms<sup>2</sup> per image for [13] and [26], whereas our full model runs at 197ms per image. All models are ResNet-50 based and timed on a single RTX 2080Ti card.

**COCO.** Results on the COCO panoptic validation set are reported in Table 3. Due to the disentangling power of our proposed pipeline and unified train-test logic, we are able to outperform the previous state-of-art method by 0.9 in terms of overall PQ, and 2.1 in terms of PQ for “stuff”.

Results on the Cityscapes test set and COCO *test-dev*

<sup>1</sup>Obtained by running our re-implementation.

<sup>2</sup>Obtained by running its publicly released code.

Method	Bb.	PQ			SQ			RQ		
		all	th.	st.	all	th.	st.	all	th.	st.
P. Deeplab [4]	R-50	58.0	—	—	—	—	—	—	—	—
Ours (w/ mask)	R-50	<b>61.0</b>	<b>52.7</b>	<b>67.1</b>	<b>81.4</b>	<b>79.6</b>	<b>82.8</b>	<b>73.9</b>	<b>66.2</b>	<b>79.6</b>
Li <i>et al.</i> [17, 1]	R-101	55.4	44.0	63.6	79.7	77.3	81.5	68.1	57.0	76.1
SSAP [8]	R-101	58.9	48.4	66.5	<b>82.4</b>	<b>82.9</b>	82.0	70.6	58.3	79.6
TASCNet [16]†	X-101	60.7	53.4	66.0	81.0	79.7	82.0	73.8	67.0	78.8
Ours (w/ mask)†	R-101	<b>63.3</b>	<b>56.0</b>	<b>68.5</b>	<b>82.4</b>	81.0	<b>83.4</b>	<b>75.9</b>	<b>69.1</b>	<b>80.9</b>

Table 4. Performance on the Cityscapes test set. Models pretrained on the COCO dataset are marked with †. Bb.: backbone, R: ResNet, X: ResNeXt.

Method	Bb.	PQ			SQ			RQ		
		all	th.	st.	all	th.	st.	all	th.	st.
JSIS-Net [6]	R-50	27.2	29.6	23.4	71.9	71.6	72.3	35.9	39.4	30.6
P. Deeplab [4]	R-50	35.2	—	—	—	—	—	—	—	—
SSAP [8]	R-50	36.9	40.1	32.0	<b>80.7</b>	<b>81.6</b>	<b>79.4</b>	44.8	48.5	39.3
TASCNet [16]	R-50	40.7	47.0	31.0	78.5	80.6	75.3	50.1	57.1	39.6
Ours (w/ mask)	R-50	<b>43.6</b>	<b>48.9</b>	<b>35.6</b>	80.1	81.3	78.3	<b>53.3</b>	<b>59.5</b>	<b>44.0</b>
Attention [18]	X-152	46.5	<b>55.9</b>	32.5	81.0	<b>83.7</b>	77.0	56.1	<b>66.3</b>	40.7
UPNet [26]	R-101	46.6	53.2	36.7	80.5	81.5	78.9	56.9	64.6	45.3
Ours (w/ mask)	R-101	<b>47.2</b>	53.5	<b>37.7</b>	<b>81.1</b>	82.3	<b>79.2</b>	<b>57.2</b>	64.3	<b>46.3</b>

Table 5. Performance on the COCO *test-dev* set. Bb.: backbone, R: ResNet, X: ResNeXt.

set are reported in Table 4 and 5. We perform *single-scale* inference, without any test-time augmentation. For fair comparison, only methods that are ResNe(X)t-based are reported. Our method achieves the state-of-art performance on both datasets with a PQ of 63.3 and 47.2 respectively.

Qualitative results are shown in Fig 7 where we compare with our re-implementation of Panoptic FPN. As the instance affinity operation integrates information from pixels locally and globally, our method can resolve errors in the detection stage by propagating meaningful information from other pixels. The “void” region (displayed in black) shown in Fig 7c are typically present in results produced by the heuristic merging process popularised by [14]. They are due to the method’s inability to resolve inconsistencies between semantic and instance predictions. In contrast, our method successfully handles such cases, as evident in Fig. 7d.

## 5. Conclusion

We have presented an end-to-end panoptic segmentation approach that exploits a novel pairwise instance affinity operation. It is lightweight, learnt from data, and capable of modelling a dynamic number of instances. By integrating information across the image in a differentiable manner, the instance affinity operation with the panoptic matching loss enables end-to-end training and heuristics-free inference, leading to improved qualities for panoptic segmentation. Furthermore, our method bestows additional flexibility upon network design, allowing our model to perform well even if it only uses bounding boxes as localisation cues.

**Acknowledgements** This work was supported by Huawei Technologies Co., Ltd., the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1. We would also like to thank the Royal Academy of Engineering and FiveAI.

## References

- [1] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 441–450, 2017. 2, 3, 4, 8
- [2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 11
- [3] Gedas Bertasius, Lorenzo Torresani, Stella X Yu, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–866, 2017. 3
- [4] 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, 2019. 7, 8
- [5] François Chollet. Xception: Deep learning with depth-wise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 7
- [6] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018. 7, 8
- [7] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 3
- [8] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 642–651, 2019. 7, 8
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 10
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.



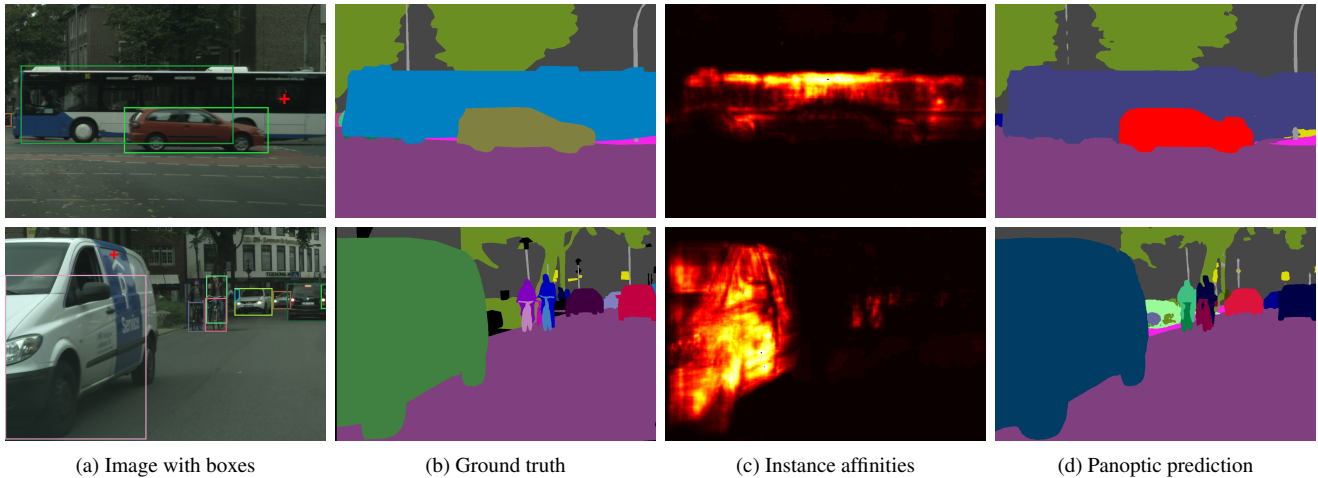


Figure 6. Examples of predicted instance affinities. The instance affinities shown in (c) are for the cross-marked pixels in (a). Observe that the predicted bounding boxes (shown in (a)) for the bus in Row 1 and the frontal car in the Row 2 fail to enclose the full object. Rule-based fusion in [14, 13] cannot recover from such localisation errors as their segments are constrained to pixels inside bounding boxes. In contrast, our model is able to still segment full objects by predicting strong affinities between the marked locations with rest of the instance.

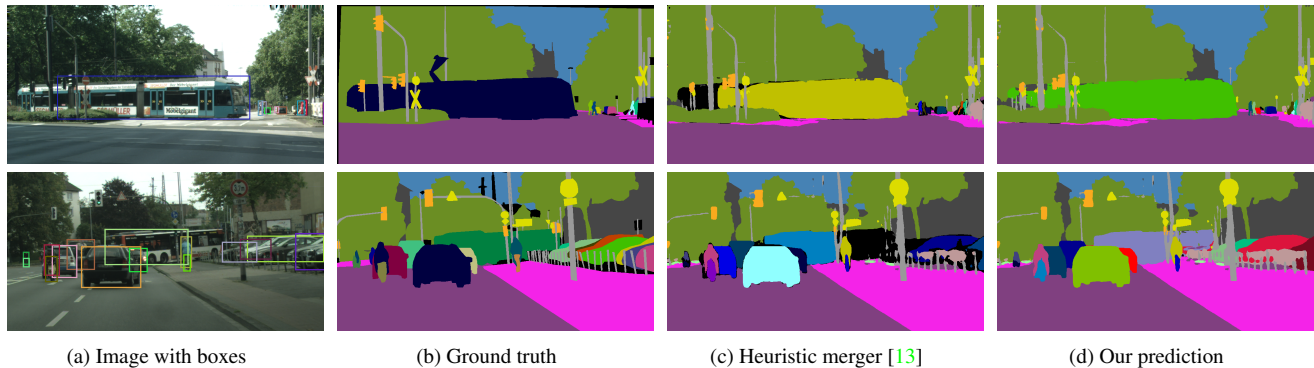


Figure 7. Qualitative results. The input images are shown with the predicted bounding boxes overlaid above. In column (c), swathes of “void” region are clearly visible for pixels where assignment cannot be made by heuristics. In contrast, our panoptic segmentation results are robust to incoherence in segmentation and localisation cues, and can explain more pixels in an image.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[12] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612, 2019. 3

[13] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vi-*

*sion and Pattern Recognition*, pages 6399–6408, 2019. 1, 2, 3, 6, 7, 9, 10, 11, 12, 14

[14] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9404–9413, 2019. 1, 2, 6, 8, 9, 12, 15

[15] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 1, 3

[16] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018. 1, 2, 7, 8, 12

- [17] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly- and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 102–118, 2018. 2, 5, 7, 8
- [18] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7026–7035, 2019. 1, 2, 7, 8, 12
- [19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3
- [20] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6181, 2019. 2, 6
- [21] Lorenzo Porzi, Samuel Rota Buló, Aleksander Colovic, and Peter Kotschieder. Seamless scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019. 7
- [22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 10
- [23] Zhuoran Shen, Mingyuan Zhang, Shuai Yi, Junjie Yan, and Haiyu Zhao. Decomposed attention: Self-attention with linear complexities. *arXiv preprint arXiv:1812.01243*, 2018. 3, 5
- [24] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 3
- [25] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3, 10, 12
- [26] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 14, 16
- [27] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deep-erlab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019. 1, 7, 12

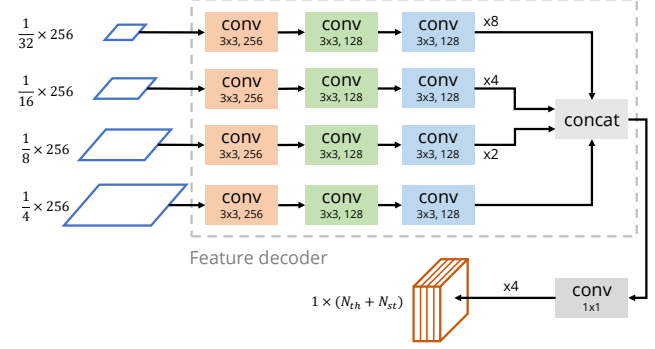


Figure H. Semantic segmentation submodule. Each  $3 \times 3$  convolution block consists of a deformable convolution (with the indicated number of output channels), a Group Normalisation operation, and a ReLU activation. Weights are **shared** across  $3 \times 3$  convolution blocks with the same colour code.

- [28] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 3
- [29] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015. 3

## Appendices

### A. Architecture and design

#### A.1 Semantic segmentation submodule

Our semantic segmentation submodule is modified from [26], by performing Group Normalisation [25] after each  $3 \times 3$  convolution. We illustrate the pipeline in Fig. H. Note that the architecture of the *feature decoder* inside this submodule is also adopted by our dense instance affinity head to extract affinity features  $Q$ . This submodule is supervised by a cross-entropy loss, unless otherwise stated.

#### A.2 Object detection submodule

In our experiments, we use the standard box head from Faster-RCNN [22] and optionally the mask head from Mask-RCNN [9] for this submodule, following [26, 13]. For the mask head, we use the Lovasz Hinge loss to replace the binary cross entropy loss. Thanks to the modular design of our network, it is easy to substitute it with any other detector architecture.

Dataset	Variant B			Variant C		
	PQ	SQ	RQ	PQ	SQ	RQ
Cityscapes	<b>61.4</b>	<b>81.8</b>	<b>74.7</b>	60.3	80.8	73.5
COCO	42.7	79.4	52.2	<b>43.4</b>	<b>79.6</b>	<b>53.0</b>

Table F. Ablation study on two design variants for the dynamic potential head. On Cityscapes, variant B outperforms variant C, whereas on COCO, variant C achieves higher accuracies.

Classified as				Classified as			
		th.	st.			th.	st.
GT	th.	95.1	4.9	GT	th.	90.1	9.9
	st.	0.0	100.0		st.	4.8	95.2
(1) Cityscapes				(2) COCO			

Table G. Confusion matrices between “thing” and “stuff” for semantic segmentation submodule outputs on Cityscapes and COCO validation sets. All numbers are percentages, normalised row-wise.

### A.3 Dynamic potential head

We refer to the design variant B and C presented in Sec. 3.4.1 (Fig. 4). At first glance, variant B, which multiplies semantic segmentation probabilities  $V_i(c_k)$  with mask scores  $M_i(k)$ , appears to be a more appropriate method than variant C which sums probabilities instead. The output of variant B is high only when both inputs are unanimously high. This can filter out spurious misclassifications from either input, and improve robustness towards false positive predictions. Indeed, on Cityscapes, we observe that variant B achieves a 1.1 PQ lead over the variant C counterpart (first row of Table F).

However, on COCO, we notice a high tendency for the semantic segmentation submodule to mistake “things” for “stuff” (Table. G2). The multiplicative action of variant B can systematically and substantially weaken the panoptic logits for “thing” classes, relative to the unattenuated panoptic logits of “stuff” classes. This can be undesirable for models whose semantic segmentation submodule is already prone to misclassifying “things” as “stuff”. On the other hand, the opposite is true for variant C, as summation strengthens panoptic logits of “things” in comparison to unmodified “stuff” scores. This led us to use variant C for COCO, and we observe a 0.7 PQ improvement in comparison to B (second row of Table F).

### A.4 Training with predicted detections

In contrast with the practice in [26], we argue that, during training, the dynamic potential head should use predicted detections instead of ground truth ones to construct its output  $\Psi$ . This allows the network to learn from realistic examples, and build up its robustness towards imperfections in

Dets. for training	PQ	SQ	RQ	IoU	$AP_{box}$
Ground truths	58.6	80.0	72.0	77.8	36.8
Predictions	<b>59.0</b>	<b>80.1</b>	<b>72.4</b>	77.8	<b>38.1</b>

Table H. Comparison between two different training strategies. The top row uses ground truth detections to train the panoptic segmentation submodule, whereas the bottom row uses the ones predicted by the network on-the-fly. Results are reported on the Cityscapes validation set.

detection localisation and scoring. To test our hypothesis, we carried out an ablation study on Cityscapes using our mask-free model. When training with ground truth boxes, a uniform score of 1.0 is used for their confidence scores. Results are shown in Table H. As expected, training with predicted detections yields performance improvements across all panoptic metrics, including a 0.4 increase in PQ. A large boost is observed for  $AP_{box}$  (+1.3), because training with predicted boxes allows gradients from the panoptic segmentation submodule to flow to the object detection submodule, giving it more fine-grained supervision. IoU has not changed, as this ablation setting does not affect the semantic segmentation module.

## B. Implementation details

**Cityscapes training.** We run our experiments on four V100-32GB GPUs. This allows us to load each GPU with eight image crops and obtain an effective batch size of 32. The large number of crops per GPU enables us to use a Lovasz Softmax loss [2] instead of a cross entropy loss for supervising semantic segmentation, which we found to be effective. Following [13], we use a base learning rate of 0.01, a weight decay of 0.0001, and train for a total of 65k iterations. The learning rate is reduced by 10 folds after the first 40k iterations, and once more after another 15k iterations. Additionally, we adopt a “warm-up” period at the start of training – linearly increasing the learning rate from a third of the base rate to the full rate in 500 iterations, which helps stabilise the training.

We augment input images on-the-fly during training to reduce the network’s tendency to overfit. Our augmentation pipeline resizes the input image by a random factor between 0.5 and 2, takes a random  $512 \times 1024$  crop, and applies a horizontal flip with 50% chance. On top of these techniques, we also apply image relighting, randomly adjusting the brightness, contrast, hue, and saturation of the image by a small amount, as used in [13].

**COCO training.** On COCO, as the dataset is larger than Cityscapes, less overfitting is observed. Therefore, in terms of data augmentation techniques, we only apply resizing where the shorter size is resized to 800 and the longer size

is kept under 1333, and random horizontal flipping with 0.5 probability.

**Miscellaneous.** We use ImageNet pretrained ResNet-50 to initialise all experiments. The batch normalisation statistics are kept unchanged, though further performance gains are likely if they are finetuned on the target dataset. When a normalisation step is used in either the semantic or panoptic submodules, we use the Group Normalisation operation [25], as it is less sensitive to batch sizes.

**Inference.** We conduct single-scale inference for all experiments, letting the network process and make predictions on full-resolution images in a single forward pass. Note that only detection predictions whose confidence scores are more than a threshold are fed into the dynamic potential head during inference, to minimise unnecessary computation. This cut-off is 0.5 and 0.75 for Cityscapes and COCO respectively.

### C. Evaluation of “stuff”

The PQ metrics effectively treats “stuff” classes as image-wide instances – making all “stuff” segments undergo the same matching procedure with ground truth segments as “thing” segments. While this approach has its merits including a unified evaluation logic and a simplified PQ implementation, it should be noted that matching “stuff” predictions to ground truth is not strictly necessary, since at most one “stuff” instance for each “stuff” class is present per image.

Furthermore, this approach towards “stuff” is neither robust nor fair as a measure for “stuff” segmentation quality, and arguably encourages post-processing of panoptic predictions. Under the PQ formulation, misclassifying even a single pixel into a “stuff” class absent in the ground truth will increment false positive detections by one, and such mistakes – exacerbated by the relatively small number of ground truth “stuff” segments in a dataset – attract a large penalty on the “stuff” RQ, even though the practical impact on perceptual quality is minimal. This also contrasts in spirit with the mean IoU metric widely adopted to measure semantic segmentation quality, as the mean IoU accumulates intersection and union counts over the whole dataset and is minimally affected by individual pixels.

On the other hand, CNN-based semantic segmentation models are typically prone to produce spurious misclassifications, as they usually do not explicitly enforce smoothness. As a result, recent panoptic segmentation works [14, 18, 16, 13, 26, 27] collectively resort to setting small “stuff” segments to “void” in the final panoptic segmentation. Therefore, to foster meaningful comparison with other state-of-the-art panoptic segmentation approaches, unless specified otherwise, we also carry out this strategy as part of evaluation.

Table I. Comparison of various evaluation metrics for “stuff”, before and after small stuff areas are set to “void” on Cityscapes validation set. Note that the  $\text{IoU}^{st}$  here is computed from the final panoptic segmentation, by combining instances of the same semantic class. This is different from the IoU metrics reported in Table 1 and 3, which measure the quality of the semantic segmentation input to the heuristic merger / our panoptic segmentation submodule.

Model	Trim stuff	PQ <sup>st</sup>	SQ <sup>st</sup>	RQ <sup>st</sup>	IoU <sup>st</sup>
Pan. FPN [13]*		59.9	79.3	72.9	74.7
Pan. FPN [13]*	✓	62.0 +2.1	79.6 +0.3	75.5 +2.6	74.5 -0.2
UPNet [26]†		60.5	79.8	73.6	75.8
UPNet [26]†	✓	62.8 +2.3	80.0 +0.2	76.3 +2.7	75.7 -0.1
Ours		64.2	81.4	77.1	78.3
Ours	✓	66.3 +2.1	81.8 +0.4	79.4 +2.3	78.2 -0.1

\* Results obtained from our re-implementation of Panoptic FPN.

† Results obtained by running the public inference script of [26].

**Effects of trimming small stuff segments on evaluation metrics.** On Cityscapes validation set, we test our full model, our re-implemented Panoptic FPN [13], and the released UPNet model [26] with and without trimming off small “stuff” regions, to quantitatively assess the impact of this step on state-of-the-art models. The findings are reported in Table I.

The results show that PQ and RQ are very sensitive to such operations, as removing small stuff segments consistently results in an increase of approximately 2 points for “stuff” PQ, and 2.5 points for “stuff” RQ. This can be largely attributed to the reduced number of false positive stuff segments. On the other hand, the “stuff” IoU metric is insensitive to such modifications, as in all three cases, it suffers a slight decrease of 0.1 or 0.2 points. This prompts us to believe that “stuff” IoU is a better metric for capturing “stuff” segmentation quality than the “thing”-centric PQ family.

### D. Detailed validation set results

We report the detailed results of our models on the Cityscapes and COCO validation sets in Table J. In addition to the metrics reported in the main paper, this table also includes breakdowns of SQ and RQ by “stuff” and “thing”.

### E. Visualisation of learnt instance affinities

Additional visualisations of some predicted instance affinities are provided in Fig. I. Note that these instance affinities are extracted from our mask-free model. Interestingly, the model has learnt to resolve cars regions covered by multiple car bounding boxes – a problem difficult for



Dataset	Method	PQ			SQ			RQ			IoU			$AP_{\text{mask}}$	$AP_{\text{box}}$
		all	th.	st.	all	th.	st.	all	th.	st.	all	th.	st.		
Cityscapes	Ours (w/o mask)	59.0	50.2	65.3	80.1	78.4	81.2	72.4	63.9	78.6	77.8	78.7	77.2	–	38.1
Cityscapes	Ours (w/ mask)	61.4	54.7	66.3	81.1	80.0	81.8	74.7	68.2	79.4	79.5	81.0	78.4	33.7	38.8
COCO	Ours (w/ mask)	43.4	48.6	35.5	79.6	80.0	78.9	53.0	59.2	43.8	53.7	60.4	43.6	36.4	40.5

Table J. Full panoptic segmentation results on Cityscapes validation set and COCO validation set. All models are ResNet-50 based, and tested with a *single-scale* inference scheme, without test-time augmentation.

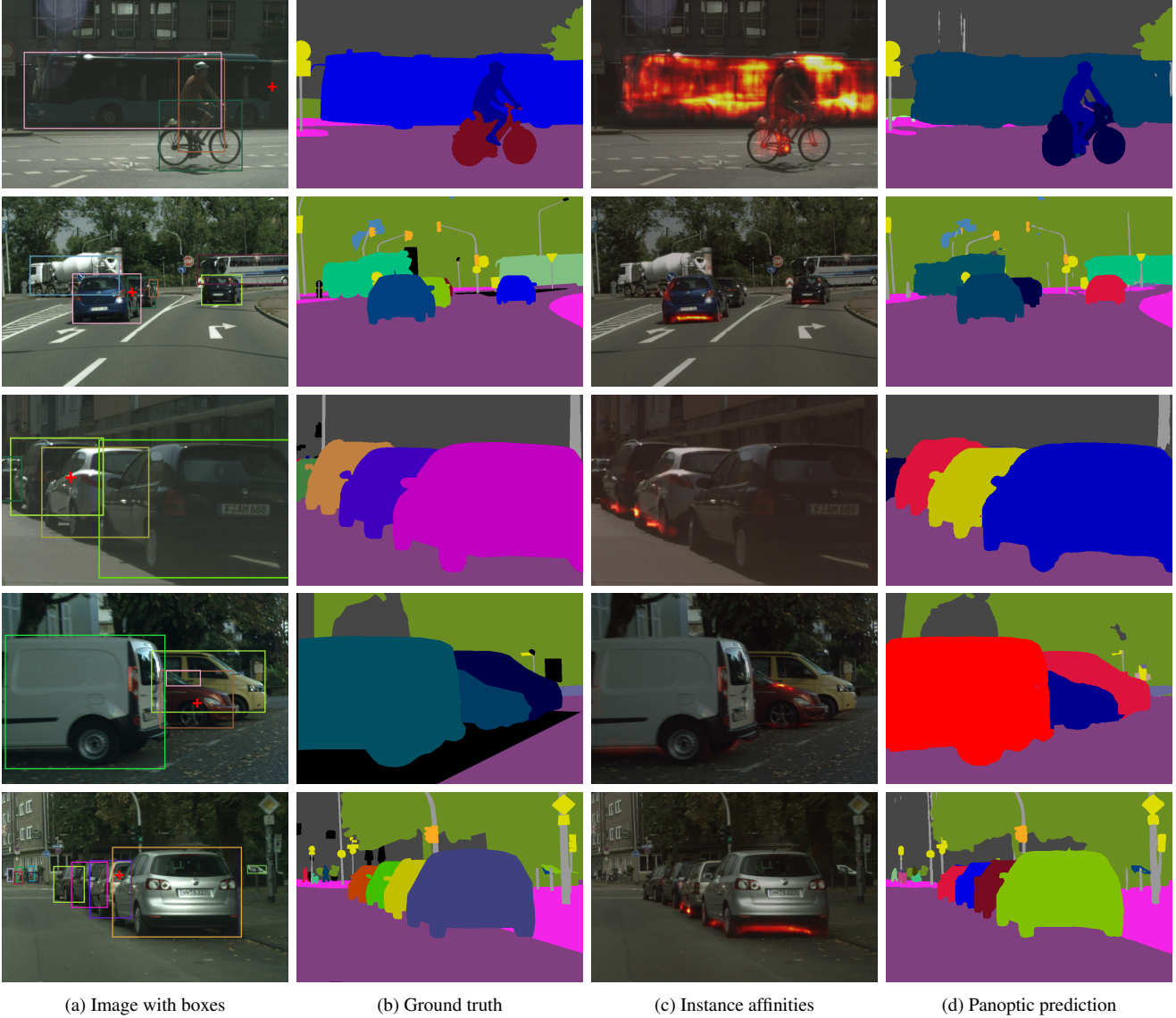


Figure I. Additional examples of instance affinities. In (c), we show the instance affinities – overlaid on input images to aid visualisation – of the cross-marked pixels in (a). These affinities and predictions are predicted by our mask-free models which use only bounding boxes. They can be seen to help segment full objects when bounding box localisation is poor (Row 1), and attribute pixels within multiple bounding boxes to the correct instances (Row 2 to 5). For Row 4, our proposed method is able to overcome a false positive detection, as the dynamic potential is robust towards false detections. For Row 5, the cross-marked pixel is on the wing mirror of the closest silver car, and our fine-grained instance affinity is able to attribute the mirror to the correct car, while the ground truth has failed to correctly label as such.

methods only using boxes as localisation cues – by creating strong instance affinities to the bottoms and tyres of cars. The model has found that these regions of cars are normally not covered by multiple bounding boxes, and therefore it is most helpful for instance discrimination by associating uncertain pixels with these regions.

## **F. Qualitative results**

We show more qualitative results in Fig. [J](#) and [K](#), and comparisons to previous state-of-the-art methods [[13](#), [26](#)].

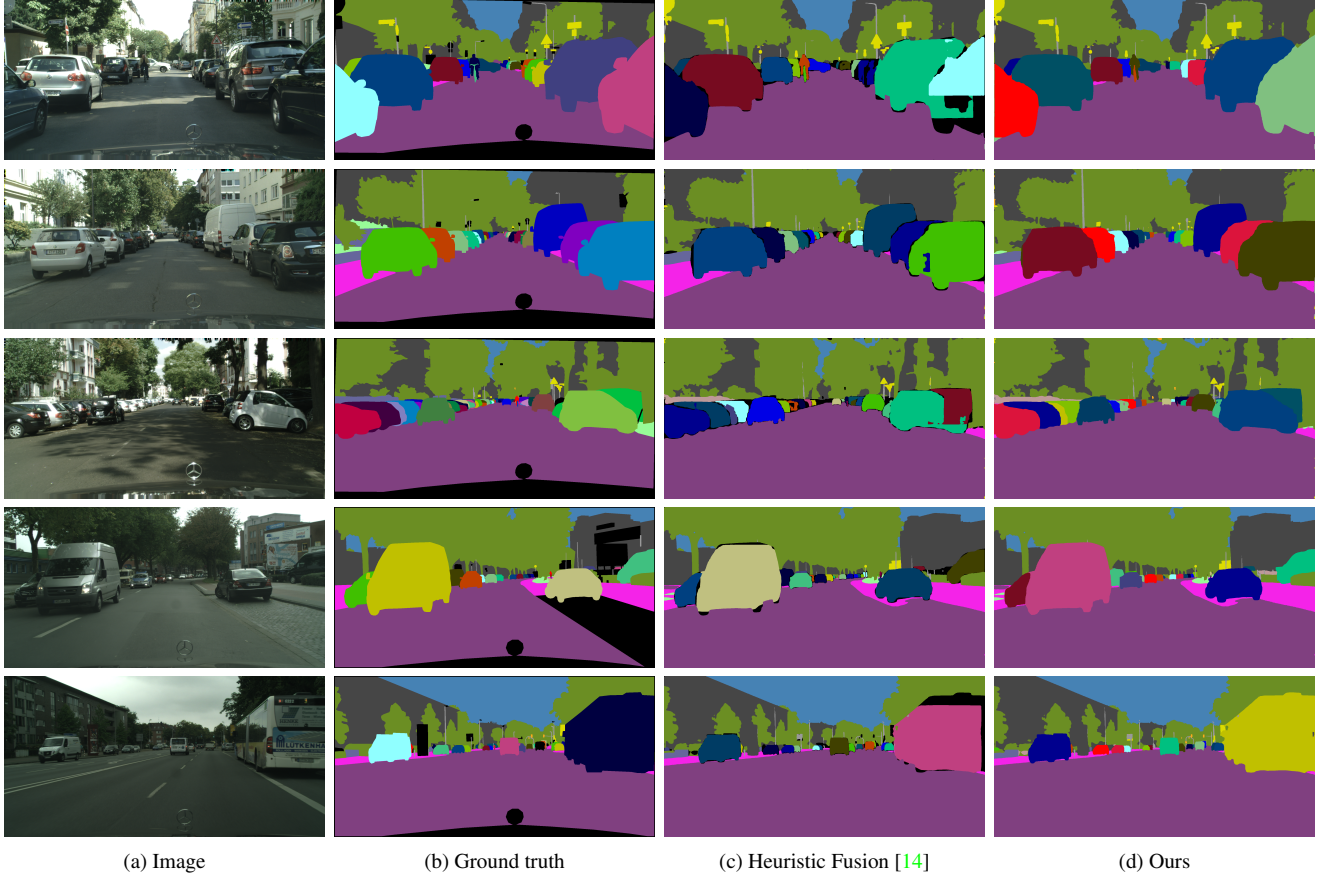


Figure J. Qualitative results on Cityscapes. Column (c) and (d) are produced by the same model under different inference strategies – either by heuristic merger [14] or with our proposed panoptic segmentation submodule. Row 1 to 3 shows that our model are able to revise erroneous cues and resolve conflicts between overlapping object masks. Row 4 and 5 demonstrate our network’s ability to segment outside boxes, when boxes do not cover the full extent of an object.



Figure K. Qualitative results on COCO. Column (c) is produced by running the publicly available inference script of [26]. With our parametrised panoptic segmentation submodule, we are able to produce more coherent, accurate, and visually appealing predictions than the parameter-free approach of [26].