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.

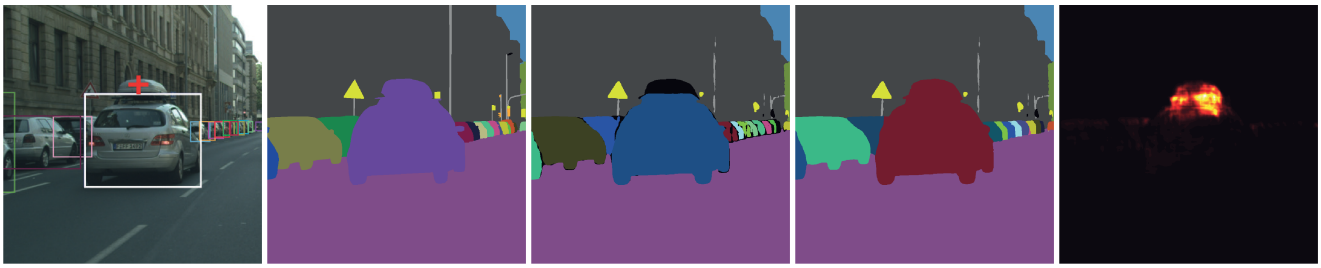
While early attempts at tackling panoptic segmentation often resort to two separate networks for instance and semantic segmentation, recent works [17, 15, 12, 24, 25] are able to improve the overall efficiency by constructing the two branches on a single, shared feature extractor, and training the multi-head, 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 [13, 12]. 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 [14, 12]. Secondly, methods without an explicit loss function for panoptic segmentation [17, 15, 12, 25] 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.



(a) Image with boxes (b) Ground truth (c) Heuristic fusion (d) Our approach (e) Instance affinities

Figure 1. Comparison of our approach *vs*. the heuristic rule-based method of [12]. 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 [12] 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 [17, 15, 12, 24]. 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 [13, 12], 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 [8] – 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 [13] 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 [15, 17]. 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 [19]. 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 [16] 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*. [24] modifies the unary terms of [1, 16] and proposes a parameter-free, differentiable panoptic head to fuse semantic and instance segmentation predictions during training. Similar to [16], 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 [14, 27, 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 [2]. Lately, another technique, self-attention, has been successful in several vision tasks [22, 26, 6]. However, its quadratic memory and computation complexity has cast doubt over its practicality. To mitigate this problem, Shen *et al*. [21] 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 [26, 11, 22], 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 [21] 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 $\boldsymbol{F}$. In our experiments, we adopt a simple ResNet-FPN backbone that outputs four multi-scale feature maps [18], following a common practice in prior works [12, 24]. To encourage global consistency, we carry out a squeeze-and-excitation operation [10] on the top-level ResNet feature before producing the first FPN feature. A similar strategy is used in [24].

3.2. Semantic segmentation submodule

The backbone features $\boldsymbol{F}$ are fed into the semantic segmentation submodule to produce a $\frac{H}{d} \times \frac{W}{d} \times (N_{st} + N_{th})$ tensor $\boldsymbol{V}$, 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 [12, 24]. In this work, we modify the design in [24] by inserting a Group Normalisation operation [23] 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 $\boldsymbol{F}$ are also passed to an object detection submodule, which generates D object detections, consisting of bounding boxes $\boldsymbol{B} = \{B_1, B_2, B_3, ..., B_D\}$, confidence scores $\boldsymbol{s} = \{s_1, s_2, s_3, ..., s_D\}$, and predicted classes $\boldsymbol{c} = \{c_1, c_2, c_3, ..., c_D\}$. 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" $\boldsymbol{M} = \{M_1, M_2, M_3, ..., M_D\}$, they are easily incorporated into the dynamic potential $\boldsymbol{\Psi}$. 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 $\boldsymbol{\Psi}$ (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.

3.4.1 Dynamic potential head

The dynamic potential head functions as an assembly node for segmentation and localisation cues from prior submod-

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.

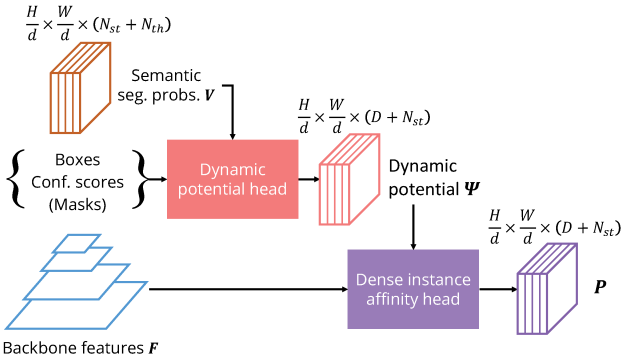


Figure 3. The panoptic segmentation submodule. Details on the dynamic potential head and dense instance affinity head are further clarified in Fig. 4 and 5 respectively.

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.

ules. 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 [24], 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 $\boldsymbol{B}$ and box classes $\boldsymbol{c}$ (including the dummy full-image "stuff" boxes), it populates each box region with a combination of semantic segmentation probabilities $\boldsymbol{V}$ and box confidence scores $\boldsymbol{s}$ to produce a *dynamic potential* $\boldsymbol{\Psi}$ with $(D + N_{st})$ channels:

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

Optionally, if provided with object masks $\boldsymbol{M}$, the dynamic potential head can also incorporate them into $\boldsymbol{\Psi}$. Defining $\boldsymbol{M}$ 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:

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

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 $\boldsymbol{\Psi}$ 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 number of input channels. This apparent incompatibility has led prior works on panoptic segmentation to use either no parameter at all [24], or only single scaling factors for entire tensors [16] providing limited modelling capacity.

This conundrum can be tackled by driving this head with a pairwise dense instance affinity, which is predicted from

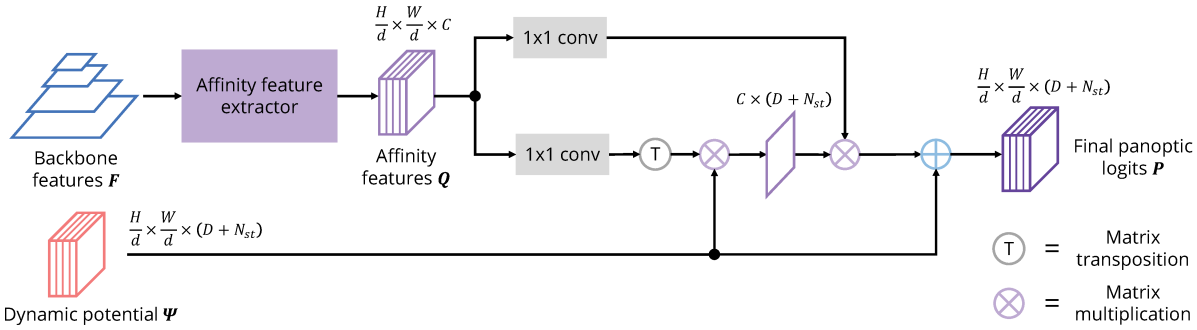


Figure 5. The dense instance affinity head. It is parametrised, expressive, lightweight, and fully differentiable.

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 `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 $\boldsymbol{F}$ a single feature tensor $\boldsymbol{Q}$ 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 $\boldsymbol{Q}$ can be easily collapsed to produce a $\frac{HW}{d^2} \times C$ feature matrix.

Normally, the pairwise instance affinities $\boldsymbol{A}$ – a large $\frac{HW}{d^2} \times \frac{HW}{d^2}$ matrix – would then be produced by performing a matrix multiplication $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{Q}^T$. This would be followed by multiplying $\boldsymbol{A}$ 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 $\boldsymbol{A}$ would occupy 15.7GB of GPU memory, making this approach unpractical.

Drawing from insight of [21], we design a lightweight pipeline for computing and applying the dense instance affinities (Fig. 5). Instead of sequentially computing $\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{\Psi}$ which explicitly produces $\boldsymbol{A}$, we compute $\boldsymbol{Q}\big(\boldsymbol{Q}^T\boldsymbol{\Psi}\big)$, since:

$$\big(\boldsymbol{Q}\boldsymbol{Q}^T\big)\boldsymbol{\Psi} = \boldsymbol{Q}\big(\boldsymbol{Q}^T\boldsymbol{\Psi}\big) \tag{3}$$

The result of $\boldsymbol{Q}^T\boldsymbol{\Psi}$ 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:

$$\boldsymbol{P} = \boldsymbol{\Psi} + \phi_0(\boldsymbol{Q})\big(\phi_1(\boldsymbol{Q}^T)\boldsymbol{\Psi}\big) \tag{4}$$

where ϕ_0 and ϕ_1 are each a 1×1 convolution followed by an activation. From this formulation, inference is straight forward and does not require any post-processing, as an `argmax` operation on $\boldsymbol{P}$ along the channel direction readily produces the panoptic segmentation prediction.

Note that we do not compute a loss directly over $\boldsymbol{Q}$; 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 $\boldsymbol{Q}$ 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 α ground truth segments $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, ..., \mathcal{T}_\alpha\}$, and a set of β predicted bounding boxes $\boldsymbol{B} = \{B_1, B_2, B_3, ..., B_\beta\}$, we find the "matched" ground truth $\boldsymbol{\mathcal{T}}^\star$ which satisfies:

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

where $\operatorname{box}(.)$ extracts tight bounding boxes from segments, $\pi(\boldsymbol{\mathcal{T}})$ refers to all permutations of $\boldsymbol{\mathcal{T}}$, 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 α and β 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 $\boldsymbol{\mathcal{T}}^{\star}$, we can compute the loss on the predicted panoptic segmentation probabilities $\boldsymbol{P}$ as per normal with a cross-entropy loss. Our experiments use 0.5 for t.

Unlike ours, the panoptic loss used by [24] does not have the matching stage and its panoptic head is trained with ground truth detections instead. As a result, the models of [24] are not trained to handle imperfect localisations. In addition, our loss differs from [19] 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) [13]. 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 $\boldsymbol{V}$, and the box Average Precision (AP_{box}) of our predicted bounding boxes $\boldsymbol{B}$. Additionally, for models which predict object instance masks $\boldsymbol{M}$ in the object detection submodule, we report its mask Average Preicision (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 [12]. 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 [12]. 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 [12], 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 $\boldsymbol{\Psi}$ is used as the final output $\boldsymbol{P}$.
- 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 [13], whereas $\mathbb{C}2$ outputs the dynamic potential $\boldsymbol{\Psi}$ as $\boldsymbol{P}$.
- 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 [12]. 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 box predictions. A similar trend is also evident with object masks enabled, between model $\mathbb{C}2$ with $\mathbb{D}2$, 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 $\mathbb{C}1$ and $\mathbb{D}1$ 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 [12]. Therefore, the 1.3 increase in PQ of $\mathbb{D}1$ over $\mathbb{C}1$ solely stems from the fact that $\mathbb{D}1$ undergoes end-to-end training, and shows that our end-to-end training strategy with the panoptic matching loss is effective.

Model	Settings msk.	aff.	e2e.	heu.	amx.	PQ all	PQ th.	PQ st.	SQ all	RQ all	IoU all	AP mask	AP box
$\mathbb{A}$					✓	54.6	46.0	60.9	77.9	68.4	75.0	–	36.9
$\mathbb{B}$		✓	✓		✓	59.0	50.2	65.3	80.1	72.4	77.8	–	38.1
$\mathbb{C}1$	✓			✓		59.3	51.4	65.0	79.8	73.2	78.1	33.8	38.1
$\mathbb{C}2$	✓				✓	59.6	52.4	64.8	80.4	72.9	78.1	33.8	38.1
$\mathbb{D}1$	✓	✓	✓	✓		60.6	52.4	66.5	80.4	74.2	79.5	33.7	38.8
$\mathbb{D}2$	✓	✓	✓		✓	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 `argmax` (amx.) of the panoptic logits (which is either $\boldsymbol{\Psi}$ for $\mathbb{A}$ and $\mathbb{C}2$, or $\boldsymbol{P}$ for $\mathbb{B}$ and $\mathbb{D}2$) or use the heuristic merging strategy [12] (heu.).

Unified training and inference pipeline. For $\mathbb{D}1$, we test a model trained end-to-end with the panoptic matching loss using the heuristic merger strategies. In contrast, for $\mathbb{D}2$, we take the same model and take `argmax` from the final panoptic logits. We can see that the $\mathbb{D}2$ still outperforms $\mathbb{D}1$ 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 [9] based except [16, 25]. 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 [16, 25], our system outperforms the previous state-of-art by 3.5 PQ, even though they utilise stronger backbones (Xception-71 [4] and ResNet-101 [9]) than ours (ResNet-50).

Speed-wise, our design compares favourably with other state-of-the-art models. On Cityscapes, inference takes 386ms[1] and 201ms[2] per image for [12] and [24], 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* 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

[1] Obtained by running our re-implementation.
[2] Obtained by running its publicly released code.

Method	PQ all	PQ th.	PQ st.	SQ all	RQ all	IoU all	AP mask	AP bbox
Li *et al.* [16]	53.8	42.5	62.1	–	–	**79.8**	–	–
DeeperLab [25]	56.5	–	–	–	–	–	–	–
SSAP [7]	58.4	**50.6**	–	–	–	–	–	–
Ours (w/o mask)	**59.0**	50.2	**65.3**	**80.1**	**72.4**	77.8	–	**38.1**
TASCNet [15]†	55.9	50.5	59.8	–	–	–	–	–
Attention [17]†	56.4	52.7	59.0	–	–	73.6	33.6	–
Pan. FPN [12]†	57.7	51.6	62.2	–	–	75.0	32.0	–
UPSNet [24]†	59.3	54.6	62.7	79.7	73.0	75.2	33.3	**39.1**
Pan. Deeplab [3]†	59.7	–	–	–	–	–	–	–
Seamless [20]†	60.3	**56.1**	63.3	–	–	77.5	33.6	–
Ours (w/ mask)†	**61.4**	54.7	**66.3**	**81.1**	**74.7**	**79.5**	**33.7**	38.8

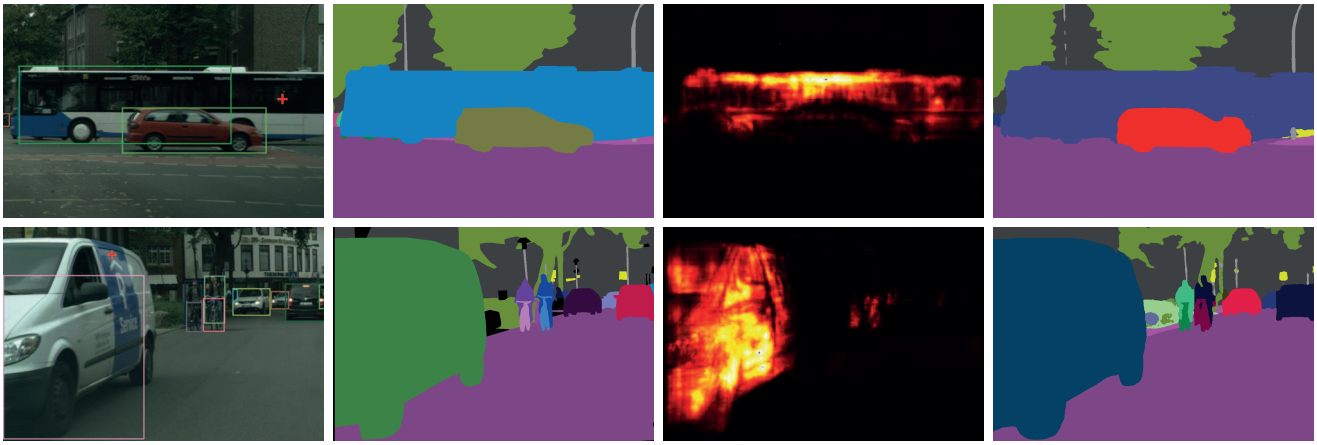
Table 2. Panoptic segmentation results on Cityscapes *val.* set. Models that run instance segmentation internally are marked with †. Other than [16, 25], all works are ResNet-50 [9] based. For fairness, we only include numbers obtained via *single-scale* inference.

Method	PQ all	PQ th.	PQ st.	SQ all	RQ all	IoU all	AP mask	AP bbox
JSIS-Net [5]	26.9	29.3	23.3	72.4	35.7	–	–	–
Pan. Deeplab [3]	35.1	–	–	–	–	–	–	–
Pan. FPN [12]	39.0	45.9	28.7	–	–	–	33.3	–
UPSNet [24]	42.5	**48.6**	33.4	78.0	52.5	**54.3**	34.3	37.8
Ours (w/ mask)	**43.4**	**48.6**	**35.5**	**79.6**	**53.0**	53.7	**36.4**	**40.5**

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

Method	Bb.	PQ all	PQ th.	PQ st.	SQ all	SQ th.	SQ st.	RQ all	RQ th.	RQ st.
P. Deeplab [3]	R-50	58.0	–	–	–	–	–	–	–	–
Ours (w/ mask)	R-50	**61.0**	**52.7**	**67.1**	**81.4**	**79.6**	**82.8**	**73.9**	**66.2**	**79.6**
Li *et al.* [16, 1]	R-101	55.4	44.0	63.6	79.7	77.3	81.5	68.1	57.0	76.1
SSAP [7]	R-101	58.9	48.4	66.5	**82.4**	**82.9**	82.0	70.6	58.3	79.6
TASCNet [15]†	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	**63.3**	**56.0**	**68.5**	**82.4**	81.0	**83.4**	**75.9**	**69.1**	**80.9**

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



(a) Image with boxes (b) Ground truth (c) Instance affinities (d) Panoptic prediction

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 [13, 12] 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.

(a) Image with boxes (b) Ground truth (c) Heuristic merger [12] (d) Our prediction

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.

Method	Bb.	PQ			SQ			RQ		
		all	th.	st.	all	th.	st.	all	th.	st.
JSIS-Net [5]	R-50	27.2	29.6	23.4	71.9	71.6	72.3	35.9	39.4	30.6
P. Deeplab [3]	R-50	35.2	–	–	–	–	–	–	–	–
SSAP [7]	R-50	36.9	40.1	32.0	**80.7**	**81.6**	**79.4**	44.8	48.5	39.3
TASCNet [15]	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	**43.6**	**48.9**	**35.6**	80.1	81.3	78.3	**53.3**	**59.5**	**44.0**
Attention [17]	X-152	46.5	**55.9**	32.5	81.0	**83.7**	77.0	56.1	**66.3**	40.7
UPSNet [24]	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	**47.2**	53.5	**37.7**	**81.1**	82.3	**79.2**	**57.2**	64.3	**46.3**

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

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 [13]. 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, 7

[2] 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

[3] 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

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 7

[5] 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

[6] 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

[7] 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

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[10] 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

[11] 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

[12] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 1, 2, 3, 6, 7, 8

[13] 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

[14] 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

[15] 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

[16] 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, 4, 7

[17] 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

[18] 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

[19] 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

[20] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019. 7

[21] 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

[22] 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

[23] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[24] 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, 6, 7, 8

[25] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019. 1, 7

[26] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 3

[27] 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