Panoptic-PartFormer: Learning a unified model for Panoptic Part Segmentation

Xiangtai Li[1], **Shilin Xu**[1], Yibo Yang[1,3], Guangliang Cheng[2], Yunhai Tong[1], Dacheng Tao[3]

[1]Peking University, [2]SenseTime Research, [3]JD Explore Academy

Paper

Code

ECCV TEL AVIV 2022

1. Motivation and Introduction

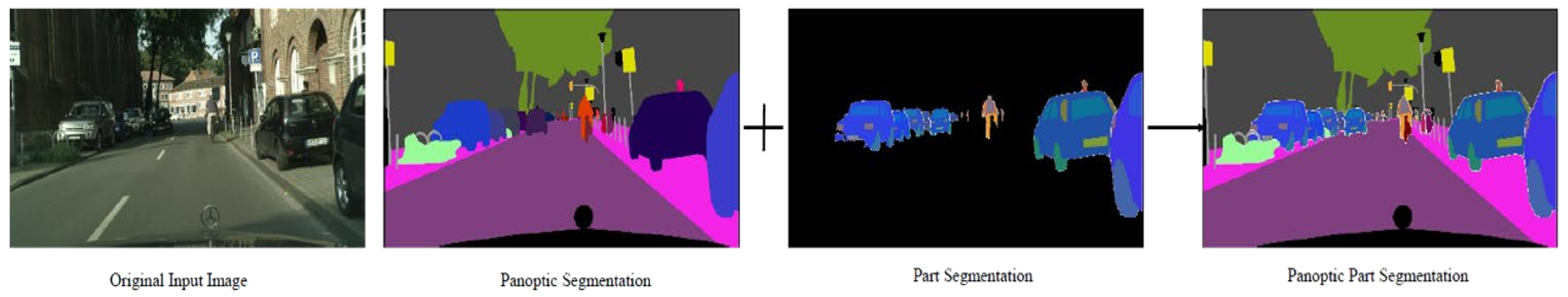


Fig.1 Part-aware Panoptic Segmentation or **Panoptic Part Segmentation (PPS)**

1.1, PPS: A new challenging task that combine Part Segmentation and Panoptic Segmentation into one **unified** framework.

It requires the model to jointly segment panoptic segmentation (**scene** level) and part segmentation (**part** level).

1.2, New metric: Part-aware Panoptic Quality (PartPQ)

$$\mathrm{PartPQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IOU}_p(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}. \quad (1) \qquad \mathrm{IOU}_p(p,g) = \begin{cases} \text{mean } \mathrm{IOU}_{\text{part}}(p,g), & l \in \mathcal{L}^{\text{parts}} \\ \mathrm{IOU}_{\text{inst}}(p,g), & l \in \mathcal{L}^{\text{no-parts}} \end{cases} \quad (2)$$

1.3 Why PPS?

1. For more fine grained scene understanding. Object parts level parsing.

2. To explore the relationship with thing part and global thing/stuff.

3. To achieve multi-level understanding of the scene.

4. Important for several application such as auto-driving.

2. Existing Methods

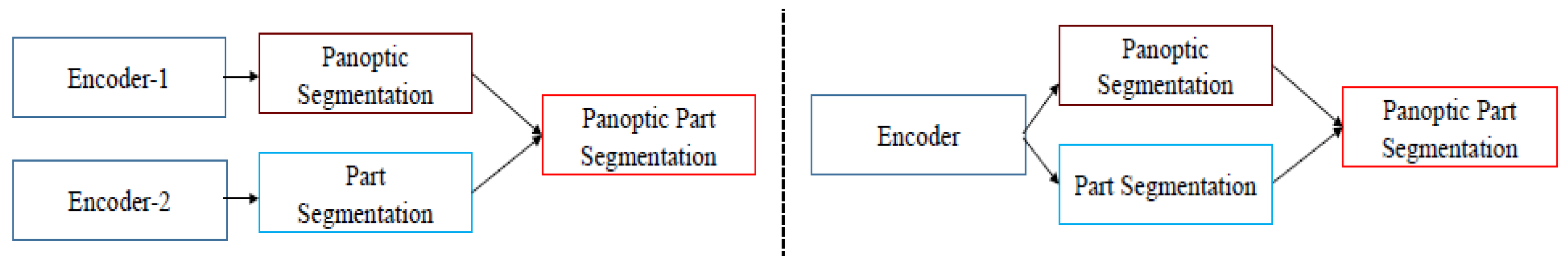


Fig. 2 (a) Different methods fusion

Fig.2 (b) Shared Backbone with Different Heads

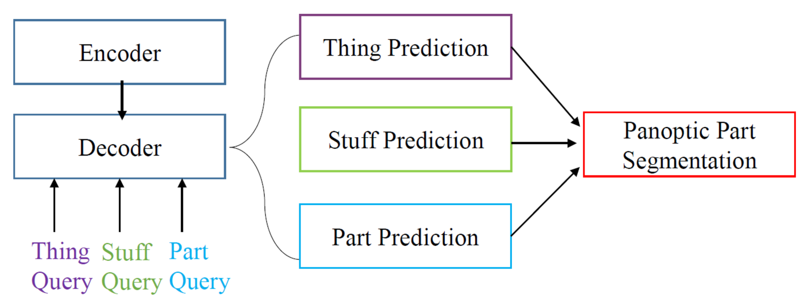
2.1 Previous Solutions:

1. Huge Computation Cost (a), Not Shared Encoder.

2. Not End-to-End Training, Complex Pipeline (a),(b)

3. No task association (a), (b). Scene and Part are independent.

4. Hard to explore the relationship between scene and part features. (b)



(c) Our Unified Solution: Panoptic-PartFormer

2.2 Our method:

1. Less Computation Cost.

2. End-to-End Training.

3. Task association via Queries.

3. Method

3.1 Key Motivation:

1. Represent thing, stuff and part as object queries in a unified format.

2. Task association can be performed via a shared transformer decoder.

3. End-to-End training and directly output multi-level masks.

4. Use Query features to jointly update the thing, stuff and part queries.

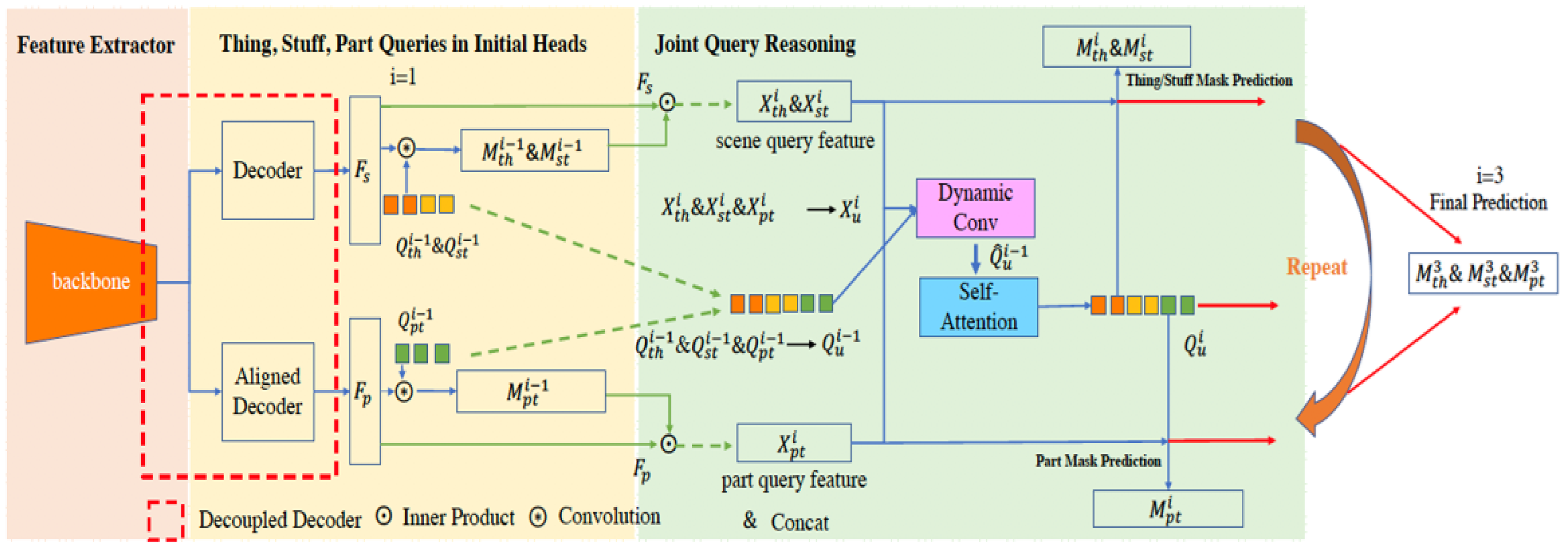


Fig.3 Panoptic PartFormer Architecture

3.2 Key Steps:

1, Feature Extractor: Backbone + Feature Pyramid Network.

2, Decoupled Decoder(DD): Decoder for scene features, Aligned Decoder for the fine grained part features.

3, Init Thing, Stuff, Part Quires on both decoder sides.

4, Perform joint query reasoning and refine the each query via a cascaded manner.

3.3 Each Stage in green region: This process is repeated three times.

1, The new query features are obtained by masked grouping in Equ.1.

$$X^i = \sum_{u}^{W}\sum_{v}^{H} M^{i-1}(u,v)\cdot F(u,v), \qquad (1)$$

2, The refined object queries are updated and weighted by the Dynamic Convolution (DC) Equ.2, Equ.3.

$$\hat{Q}_u^{i-1} = DynamicConv(X_u^i, Q_u^{i-1}), \qquad (2)$$

$$\hat{Q}_u^{i-1} = Gate_x(X_u^i)X_u^i + Gate_q(X_u^i)Q_u^{i-1}, \qquad (3)$$

3, Thing, Stuff, Part queries are reasoned by Self-Attention (SA) jointly Equ.4.

$$Q_u^i = FFN(MHSA(\hat{Q}_u^{i-1}) + \hat{Q}_u^{i-1}), \qquad (4)$$

3.4 Loss Function and Inference.

1, Mask based Cross Entropy Loss and Dice Loss.

2, Directly Output the Thing, Stuff, Part masks in one framework.

4. Experiments

Panoptic seg. method	Part seg. method	PQ All	PQ P	PQ NP	PartPQ All	PartPQ P	PartPQ NP
Cityscapes Panoptic Parts validation set							
UPSNet [64](ResNet50)	DeepLabv3+ [3](ResNet50)	59.1	57.3	59.7	55.1	42.3	59.7
DeepLabv3+(ResNet50) & Mask R-CNN(ResNet50) [18]	DeepLabv3+ [3] (Xception- 65)	61.0	58.7	61.9	56.9	43.0	61.9
Panoptic-PartFormer (ResNet50)		**61.6**	**60.0**	**62.2**	**57.4**	**43.9**	**62.2**
EfficientPS [45](EfficientNet) [53]	BSANet [75](ResNet101)	65.0	64.2	65.2	60.2	**46.1**	65.2
HRNet-OCR (HRNetv2-W48) [70,59] & PolyTransform [32]	BSANet [75](ResNet101)	66.2	64.2	67.0	61.4	45.8	67.0
Panoptic-PartFormer (Swin-base)		**66.6**	**65.1**	**67.2**	**61.9**	**45.6**	**68.0**

Tab.1 Results on Cityscapes Panoptic Parts Dataset

Method	PQ	PartPQ	Param(M)	GFlops
UPSNet + DeepLabv3+ (ResNet50)	59.1	55.1	>87	>890
Panoptic-PartFormer (ResNet50)	**61.6**	**57.4**	**37.35**	**185.84**
HRNet(OCR) +PolyTransform + BSANet	66.2	61.4	>181	>1154
Panoptic-PartFormer (Swin-base)	**66.6**	**61.9**	**100.32**	**408.52**

Tab.2 Detailed Comparison

Panoptic seg. method	Part seg. method	PQ	PartPQ
***Pascal Panoptic Parts** validation set*			
DeepLabv3+ & Mask R-CNN [18](ResNet50)	DeepLabv3+ [3](ResNet50)	35.0	31.4
DLv3-ResNeSt269 [71] & DetectoRS [49]	BSANet [75]	42.0	38.3
Our Unified Approach			
Panoptic PartFormer (ResNet50)		**47.6**	**37.8**
Panoptic PartFormer (ResNet101)		**49.2**	**39.3**

Tab.3 Results on Pascal Panoptic Parts Dataset

Experiments Results:

1. New STOA results on CPP and PPP datasets. (Tab.1 and Tab.3)

2. Less Parameters and Gflops (Tab.2) but Better Performance.

DD	DC	SA	I=1	I=3	PQ	PartPQ
✓	✓	✓	-	✓	61.6	57.4
-	✓	✓	-	✓	61.2	55.9
✓	-	✓	-	✓	57.0	52.2
✓	✓	-	-	✓	57.3	53.4
✓	✓	✓	✓	-	58.3	54.2

(a) Effect of each component. DD: Decoupled Decoder. DC: Dynamic Convolution. SA: Self Attention. I: Interaction number.

Setting	PQ	PartPQ
Joint Reasoning	61.6	57.4
Separate Reasoning	61.1	56.8
Sequential Reasoning	60.8	56.3

(b) Ablation on Query Reasoning Design

Ablation Study:

(a) Effect of Each Components.

(b) Effect of Query Reasoning Design

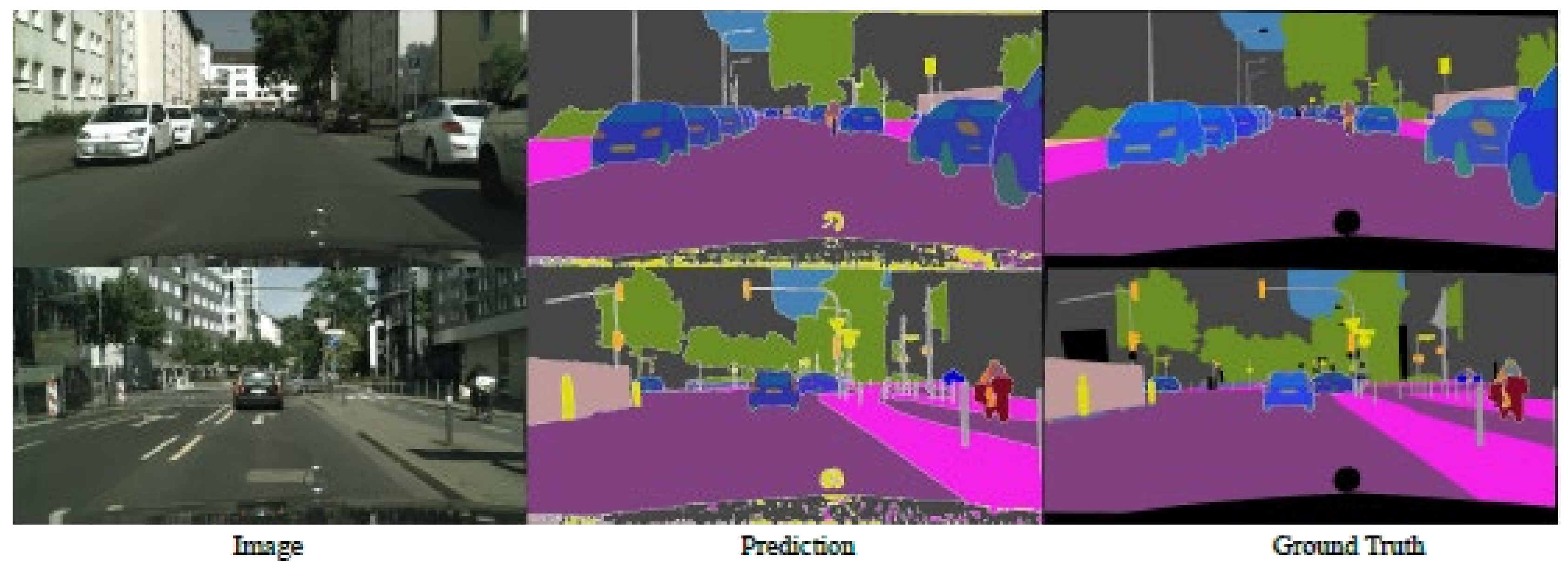


Fig3. Visualization on Cityscape Panoptic Part

Fig4. Visualization on Pascal Context Panoptic Part