

ParNeC

模式识别与神经计算研究组

PATtern Recognition and NEural Computing



面向高效推理的半监督目标检测方法研究

欣子豪

大纲

01 现有工作

02 研究框架

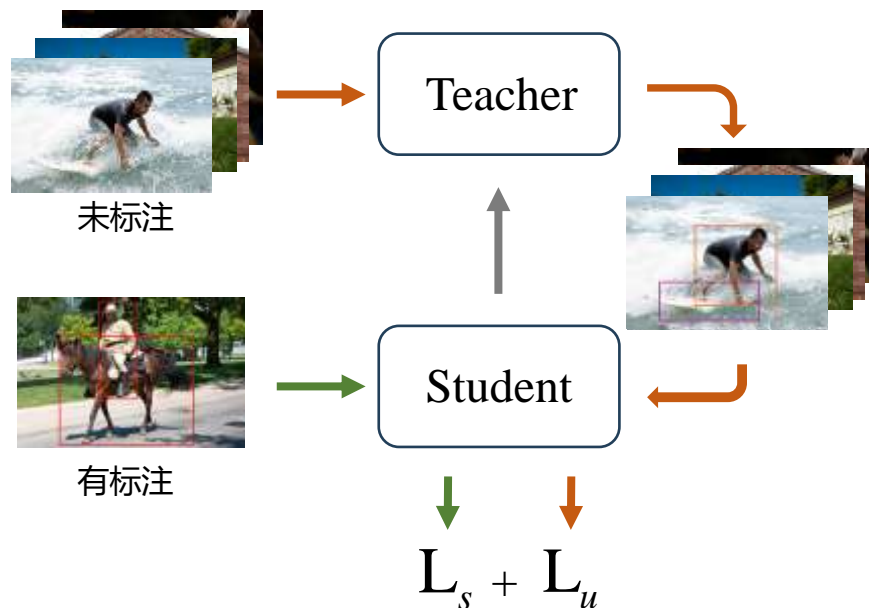
03 Smart Teacher

04 Efficient Semi-DETR

01

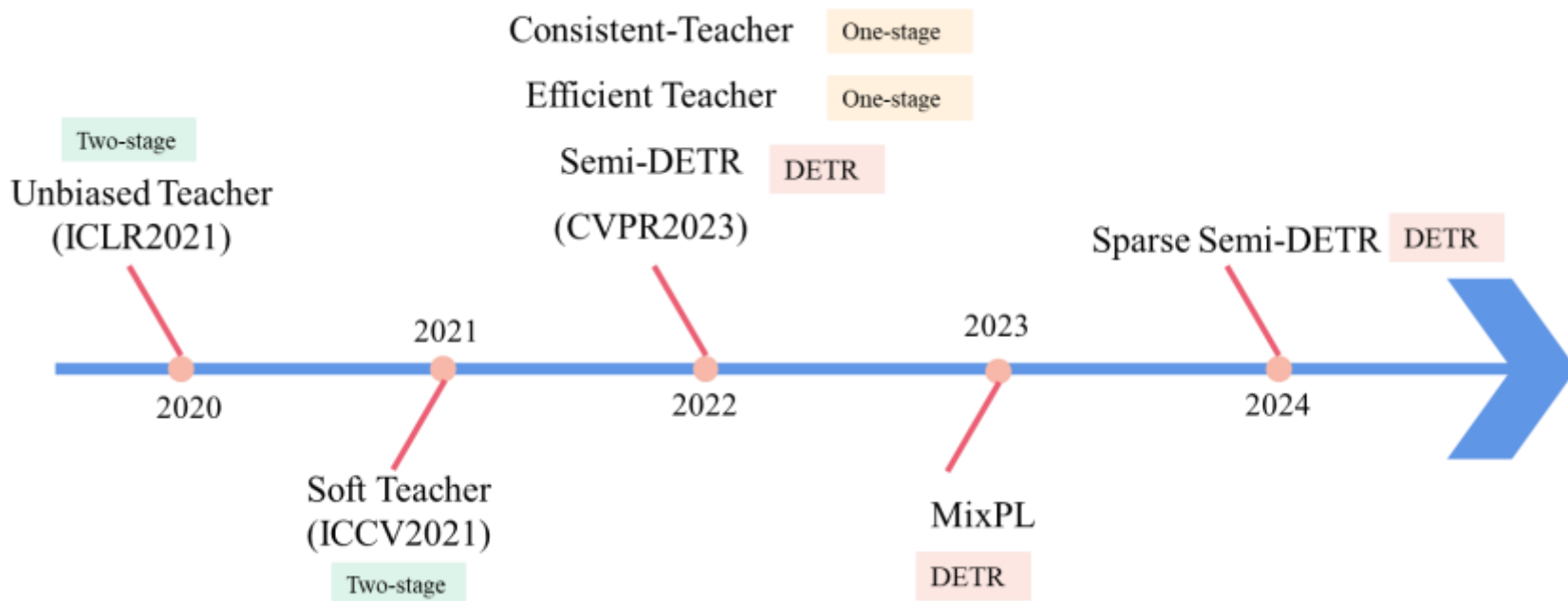
—— 现有工作 ——

半监督目标检测



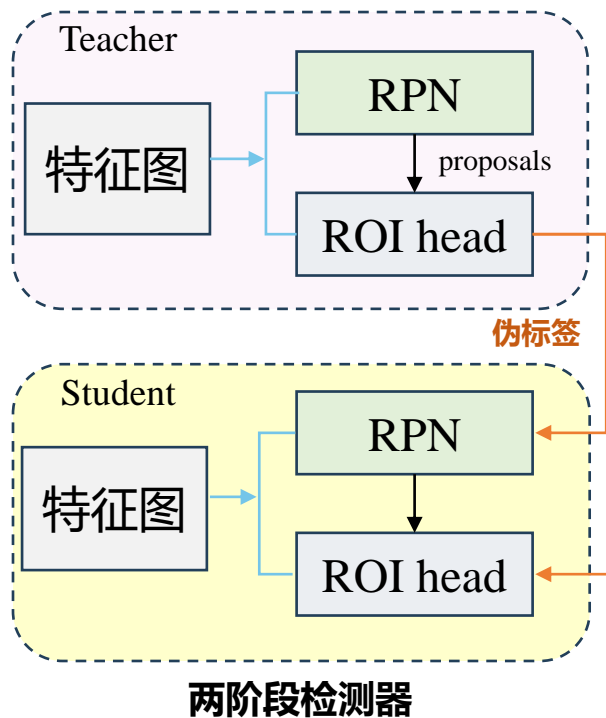
充分利用**有限的标注数据**和丰富的**未标注数据**，
以提升模型性能并降低对标注数据的依赖。

现有工作

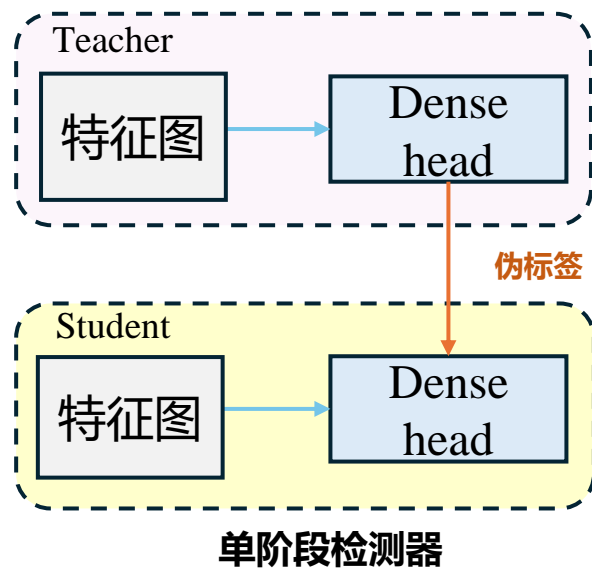


根据基础检测器可以分为两阶段、单阶段和DETR三类。

现有工作

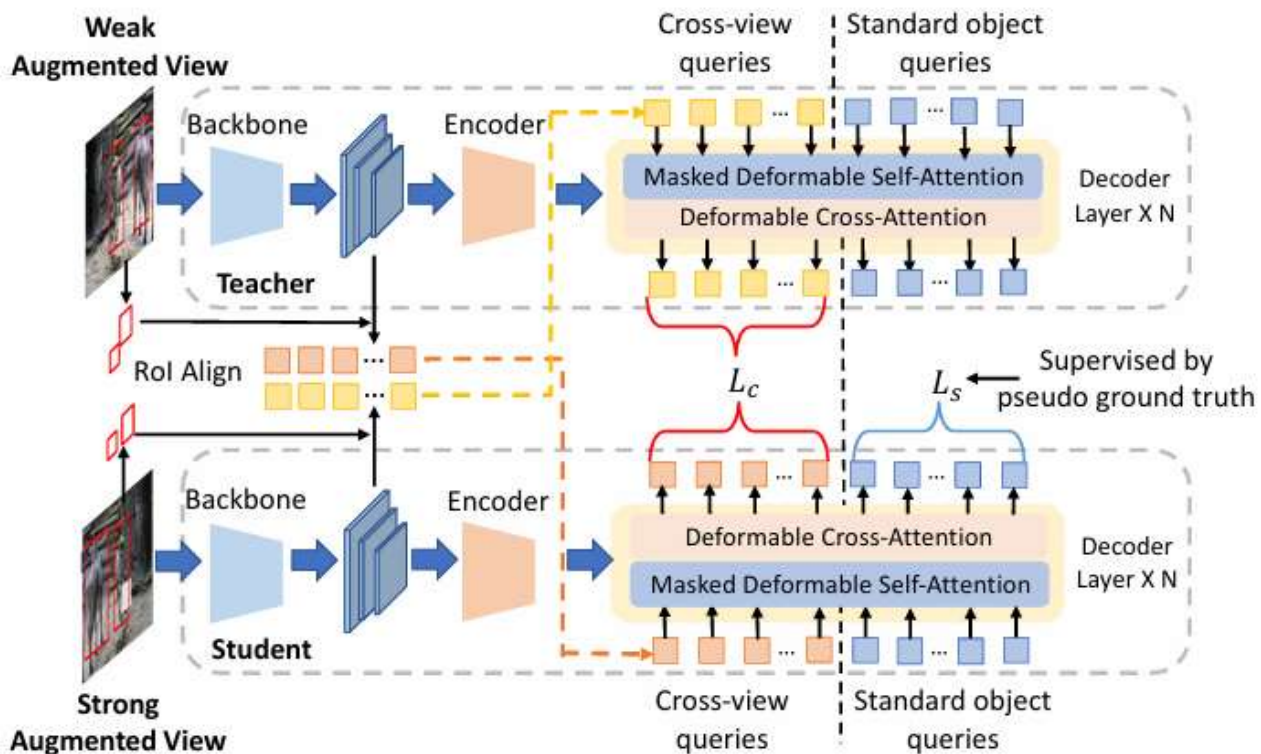


✘ RPN模块通过回归框抖动等策略能够筛选鲁棒的伪标签，但是并行计算效率差



✘ 严重正负样本不平衡问题，并需要通过额外模块提高伪标签质量，增加计算量。

现有工作



Semi-DETR[1] 训练大约需要280G显存, 在八卡A100上需要20+小时。

现有工作

✘ 在轻量检测器上失效

现有方法需要**足够数量的高质量伪标签**，通过缩减参数规模会导致半监督方法失效，低质量的伪标签会在训练过程中过程中被迭代放大，形成误差雪崩效应。

Model	AP50:95	AP50
Supervised(yolov5l)	29.8	45.9
EfficientTeacher	37.9 (+8.1)	54.2 (+8.3)
Supervised(yolov5s)	23.2	39.2
EfficientTeacher(yolov5s)	25.0 (+1.8)	40.8 (+1.6)

✘ 存在问题

1) 缩小现有半监督检测模型参数规模只减少了模型的理论计算量但模型整体**并行计算效率不高**，不能够显著提高模型推理性能。

2) 轻量检测器**伪标签质量差**，无法有效指导学生模型；

3) 两阶段和DETR检测器**计算复杂度高**，单阶段检测器存在严重**正负样本不平衡**问题，在轻量检测器上尤为严重。

伪标签评估



(a)



(b)



(c)



(d)

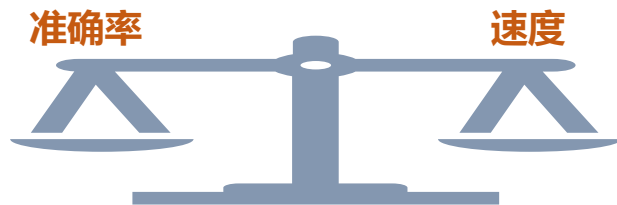


现有方法通过额外的objectness分支评估伪标签的不确定性，引入额外计算开销的同时无法真实反映伪标签质量

02

研究框架

研究框架



高效推理半监督目标检测算法



伪标签

评估伪标签质量
伪标签筛选策略

基于轻量检测器的半监督目标检测算法

- 候选伪标签特征挖掘
- 长尾分布和复杂场景下的高效检测



计算效率优化

边缘设备部署
计算与通信瓶颈

半监督目标检测算法框架优化

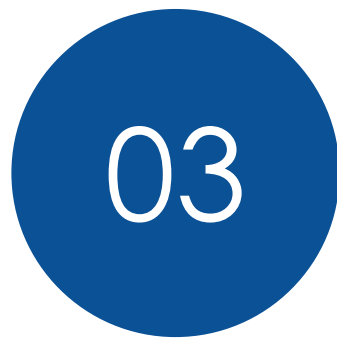
- 基于在线异构的半监督检测框架
- 并行效率与通信优化



主动标注策略

高效主动采样的半监督目标检测算法

- 基于多实例图片的主动采样方法
- 单次主动采样与模型微调



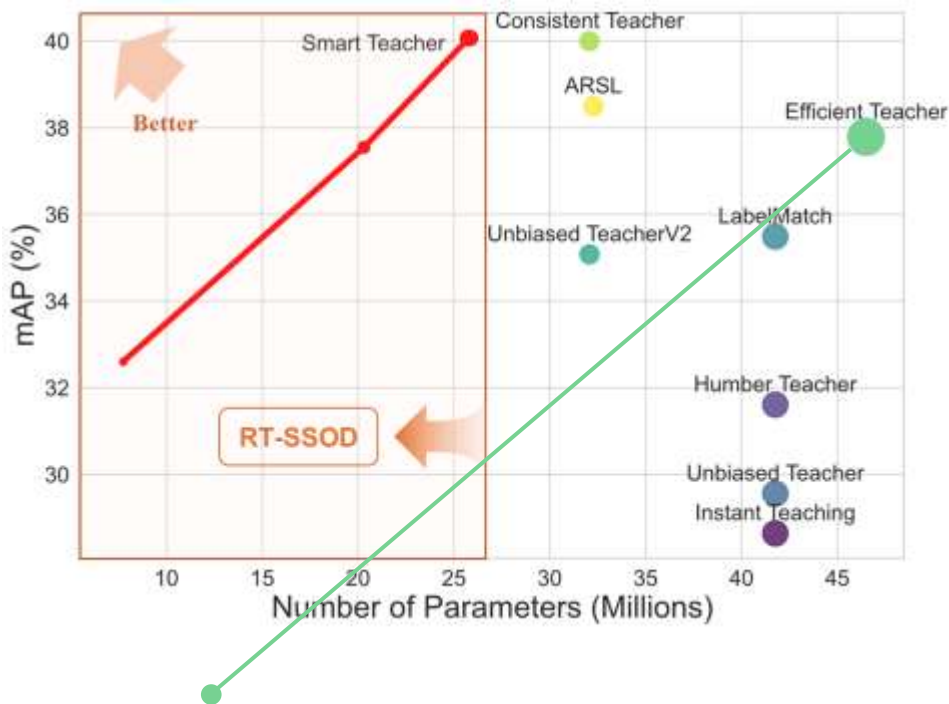
Smart Teacher

Efficient Semi-DETR

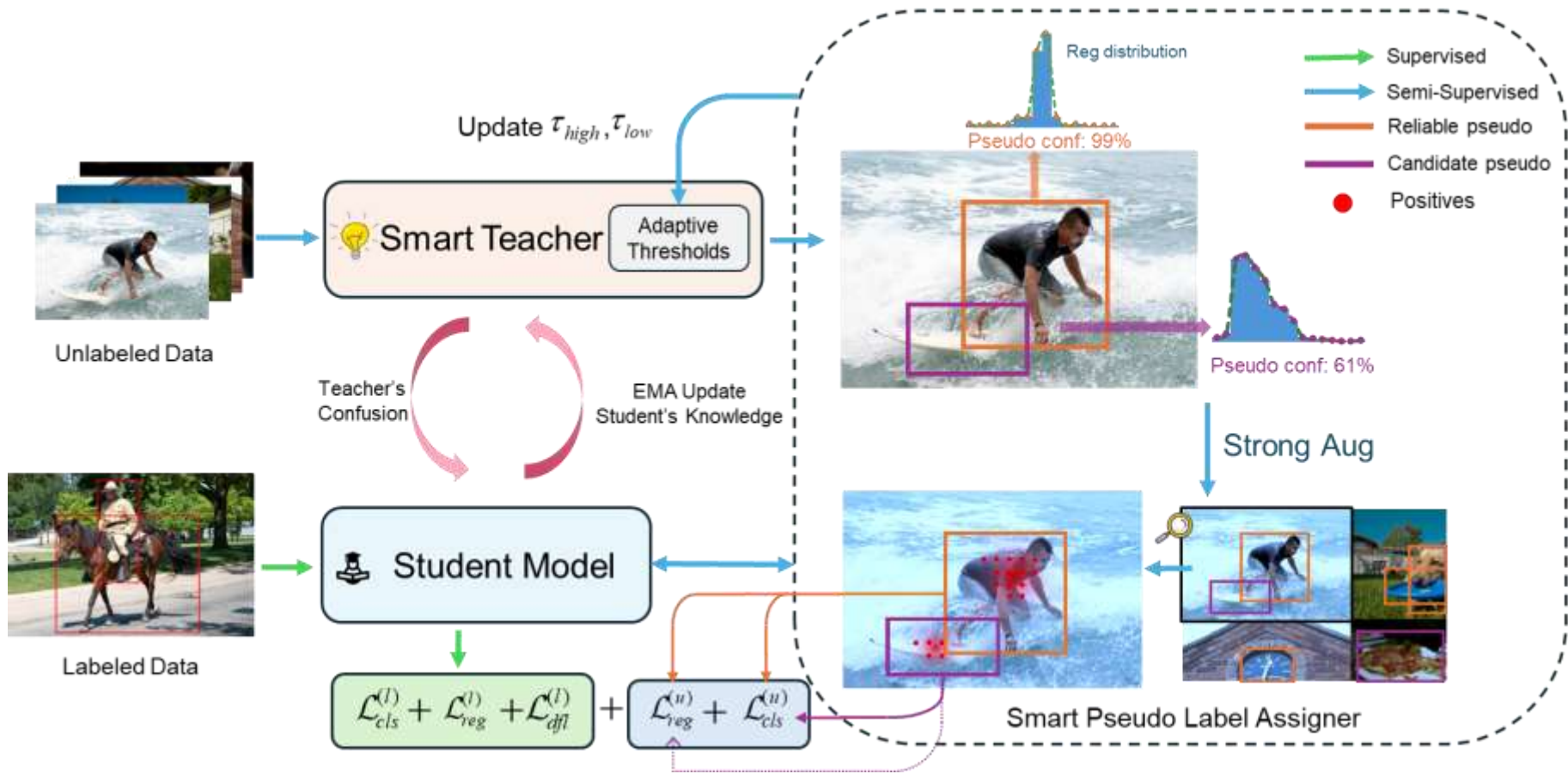
Smart Teacher: 主要贡献

实时半监督目标检测算法

- ✓ 首个基于轻量检测器的**实时**半监督目标检测算法;
- ✓ 提出了新型**伪标签评估策略**, 在不增加推理开销的情况下高效评估伪标签质量;
- ✓ 采用**教师-学生动态学习机制**, 教师模型根据不同类别的学习进展动态调整学习策略, 以及教师-学生互学习机制。



Smart Teacher

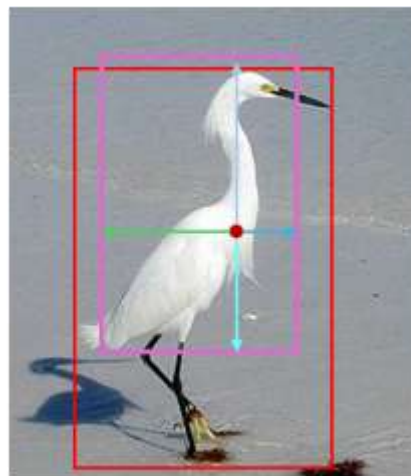


伪标签评估

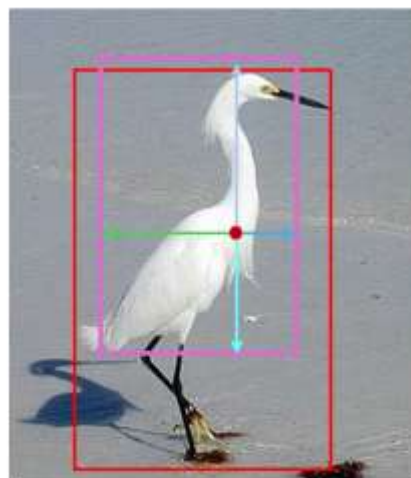
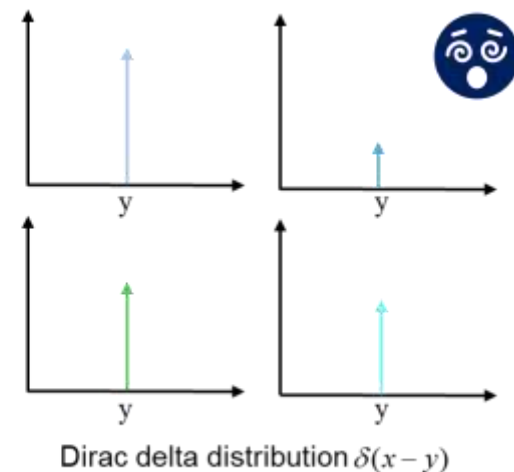
★ 针对**每个回归框精确评估**伪标签质量，而不是训练额外分支；

★ 不引入额外的计算开销；

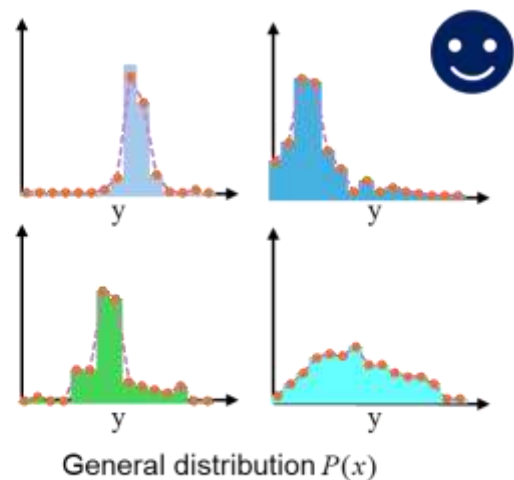
★ 将回归转化为概率分布预测，同时提供更加平滑的梯度。



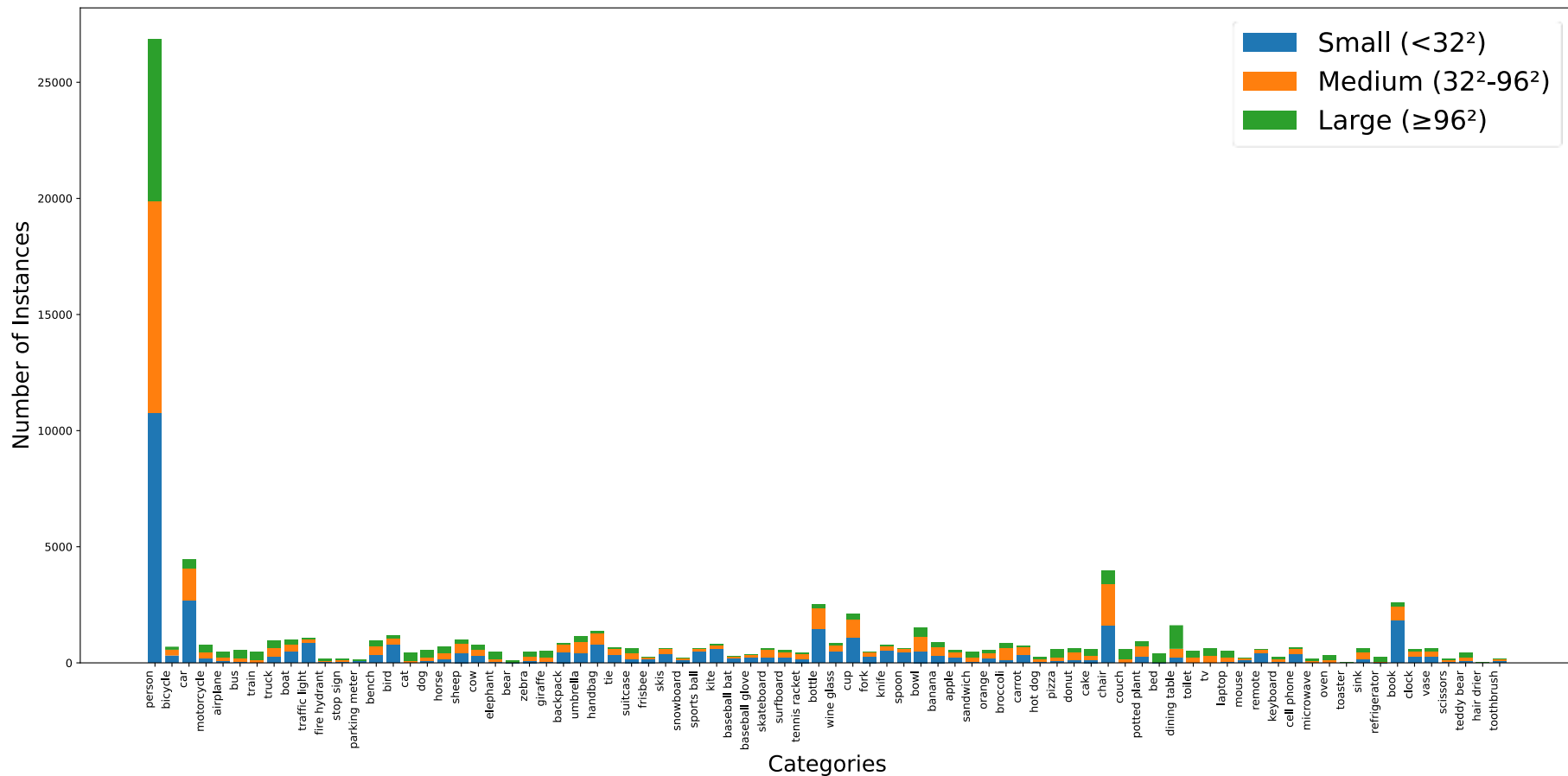
cls score: 0.91



cls score: 0.91
pseudo conf: 0.72



长尾分布问题



COCO数据集存在严重长尾分布问题，且不同类别物体的大小分布存在差异。

长尾分布问题

- 分而治之策略:

$$\begin{aligned}\mathcal{L}_u = & \sum_i^{N_u} [\mathbb{I}(p_t(x_i^u|\omega) \geq \tau_{low}) \mathcal{L}_{cls}(\Omega(x_i^u), \hat{y}_i^u) \\ & + \mathbb{I}(p_t(x_i^u|\omega) < \tau_{low}) \mathcal{L}_{cls}(\Omega(x_i^u), \mathbf{0}) \\ & + \mathbb{I}(p_t(x_i^u|\omega) \geq \tau_{high}) \mathcal{L}_{reg}(\Omega(x_i^u), \hat{y}_i^u)]\end{aligned}$$

低质量的候选伪标签只计算简单的分类损失

可靠伪标签计算分类和回归损失

长尾分布问题

- 动态学习策略:

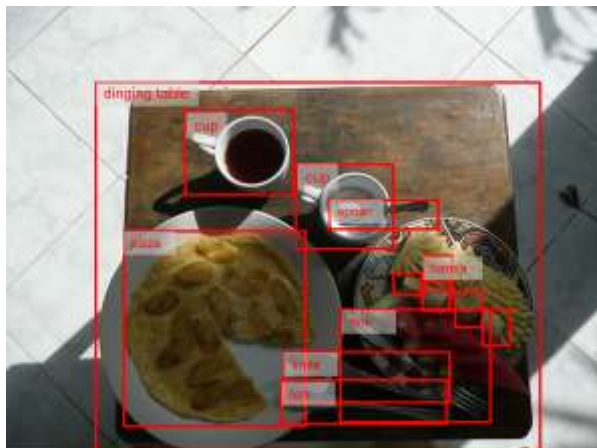
$$\begin{cases} \tau_{high,k+1}^c = \lambda_h \tau_{high,k}^c + (1 - \lambda_h) P_k^c(\lceil \gamma_1 \cdot n_k^c \rceil), \\ \tau_{low,k+1}^c = \max(0.1, P_k^c(\lceil \gamma_2 \cdot n_k^c \rceil) - \text{std}(P_k^c)). \end{cases}$$

基于最近的伪标签置信度分布更新类阈值

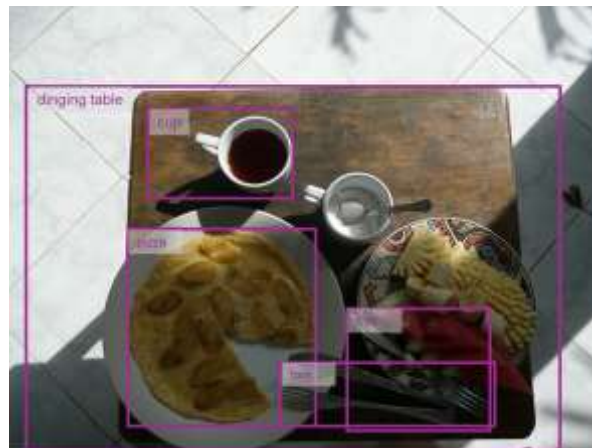
- 互学习策略:

教师-学生同时对不同扰动输出有一致性输出，视为可靠伪标签。

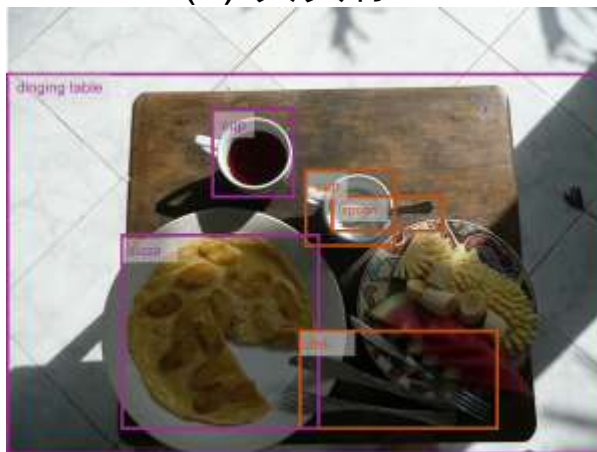
主要贡献



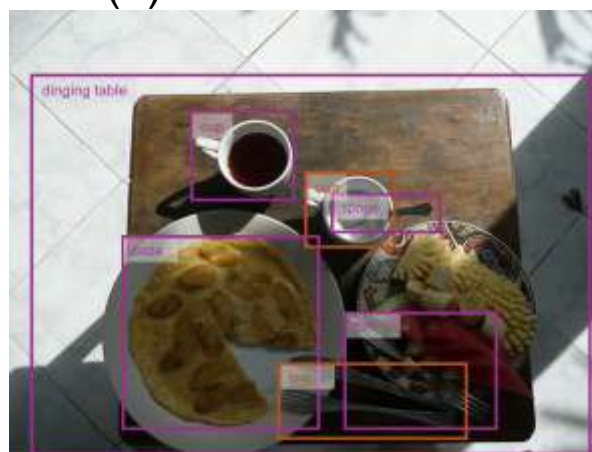
(a) 真实标签



(b) ConsistTeacher



(c) SmartTeacher-S



(d) SmartTeacher-L



候选伪标签



可靠伪标签

实验结果

Method		1%	2%	5%	10%	Params(M)
Two-stage	Faster RCNN(supervised)	10.02 ± 0.38	15.04 ± 0.31	20.82 ± 0.13	26.44 ± 0.11	41.76
	STAC	13.97 ± 0.35	18.25 ± 0.25	24.38 ± 0.12	28.64 ± 0.21	41.76
	Instant Teaching	18.05 ± 0.15	22.45 ± 0.15	26.75 ± 0.05	30.40 ± 0.05	41.76
	Humble Teacher	16.96 ± 0.38	21.72 ± 0.24	27.70 ± 0.15	31.61 ± 0.28	41.76
	Unbiased Teacher	18.33 ± 0.19	22.23 ± 0.21	26.65 ± 0.31	29.56 ± 0.24	41.76
	Soft Teacher	20.46 ± 0.39	24.89 ± 0.26	30.74 ± 0.08	34.04 ± 0.14	41.76
	LabelMatch	25.81 ± 0.28	28.56 ± 0.19	32.70 ± 0.18	35.49 ± 0.17	41.76
	PseCo	22.43 ± 0.36	27.77 ± 0.18	32.50 ± 0.08	36.06 ± 0.21	41.76
One-stage	FCOS(supervised)	9.05 ± 0.31	14.40 ± 0.28	20.69 ± 0.22	26.01 ± 0.15	32.27
	FCOS(supervised)*	3.56 ± 0.25	8.35 ± 0.30	13.90 ± 0.15	18.90 ± 0.22	32.27
	Unbiased Teacher V2	22.25 ± 0.42	26.03 ± 0.12	30.08 ± 0.04	32.61 ± 0.03	32.27
	DSL	22.03 ± 0.28	25.19 ± 0.37	30.87 ± 0.24	36.22 ± 0.18	32.27
	ARSL	25.36 ± 0.32	29.08 ± 0.21	34.41 ± 0.16	38.50 ± 0.05	32.27
	Efficient Teacher*	23.76 ± 0.13	28.70 ± 0.14	34.11 ± 0.09	37.90 ± 0.04	46.50
	Consistent Teacher	25.30	30.40	36.10	40.00	32.07
	Consistent Teacher*	19.02 ± 0.36	23.56 ± 0.31	29.02 ± 0.19	33.08 ± 0.26	32.07
	Smart Teacher-S*	15.13 ± 0.33	21.25 ± 0.22	26.21 ± 0.29	32.51 ± 0.23	7.5
	Smart Teacher-M*	19.86 ± 0.38	25.93 ± 0.27	32.95 ± 0.23	37.55 ± 0.11	20.3
	Smart Teacher-L*	23.10 ± 0.42	28.67 ± 0.35	35.29 ± 0.27	40.08 ± 0.16	25.8

10% COCO数据集实验结果

实验：推理延迟

Method	FPS _{bs=1}	Latency(ms)
ConsistentTeacher	29.6	33.8
ARSL	37.7	26.5
Efficient Teacher	59.3	16.9
Smart Teacher-S	116.9	8.6
Smart Teacher-M	100.7	9.9
Smart Teacher-L	92.6	10.8

RTX3090

Device	Scale	FPS	Latency(ms)
AGX Orin (32GB)	small	198	5.1
	medium	106	9.4
	large	82	12.2
Orin Nano (8GB)	small	91	11.0
	medium	49	20.4
	large	37	27.0

边缘设备 (TensorRT)

算法优势：

- ✓ 在不引入额外推理开销的情况下将半监督学习应用于轻量目标检测器；
- ✓ 模型仅包含卷积等简单算子，可以轻松通过算子融合实现推理加速！

边缘设备部署挑战：

- ★ 采用CPU和GPU共享内存的SoC架构，存在内存带宽竞争，且只能部署小参数量模型；
- ★ 缺乏高维张量分解、自定义插值、动态Shape等复杂运算的硬件支持；
- ★ 大量条件判断和分支跳转容易导致线程发散；

实验：推理延迟

Model	$AP_{50:95}$	AP_{50}
YOLOv5-L		
Supervised	29.81	45.92
ET	37.90(+8.09)	54.19(+8.27)
Ours	38.58(+8.77)	54.62(+8.70)
YOLOv5-S		
Supervised	23.22	39.25
ET	25.01(+1.79)	40.83(+1.58)
Ours	30.52(+7.30)	46.26(+7.01)

Efficient Teacher对比试验

Detector	$AP_{50:95}$	AP_{50}
YOLOv5-S	30.52	46.26
YOLOv8-S	32.96	46.92
YOLOv9-S	32.51	46.63
YOLOv10-S	29.63	39.66
YOLOv5-M	36.50	49.96
YOLOv8-M	37.21	51.30
YOLOv9-M	37.55	51.92
YOLOv10-M	36.65	50.01
YOLOv5-L	38.58	54.62
YOLOv8-L	39.97	55.36
YOLOv9-C	40.08	55.22

适用于不同基础检测器

算法优势：

✓ 首个适用于轻量检测器的半监督目标检测方法；

✓ 即插即用：该方法能够适用于各种检测器！

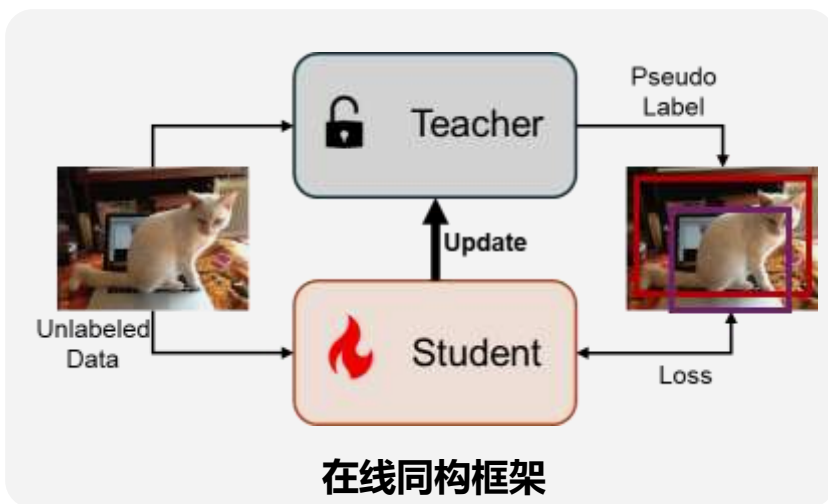
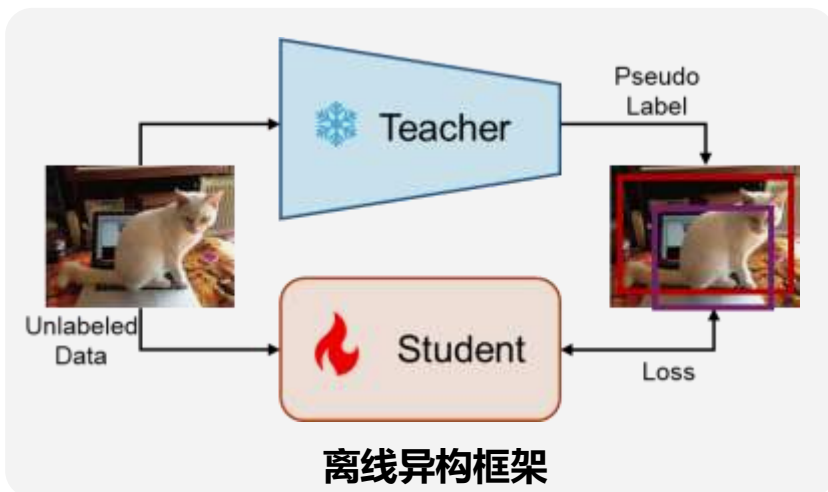
全监督检测器的提升适用于半监督任务吗？

Efficient Semi-DETR: 主要贡献

基于DETR的高效半监督目标检测算法

- ✓ 提出了在线异构教师-学生框架;
- ✓ 显著降低基于DETR半监督目标检测算法的训练成本, 推理速度是Semi-DETR的五倍;
- ✓ 采用高效的两阶段查询匹配策略, 不引入推理开销的同时解决了现有的重复检测问题。

Efficient Semi-DETR



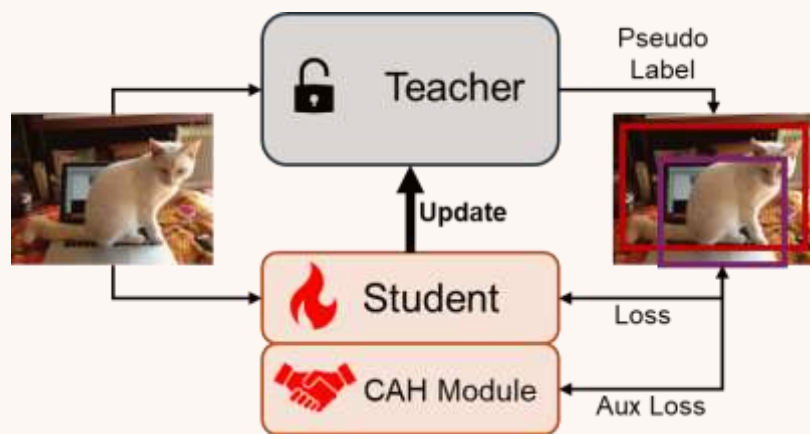
离线框架缺陷:

- ✗ 缺乏有效的动态更新机制;
- ✗ 学生模型会错误学习教师模型偏见;
- ✗ 学习效率低;

同构框架缺陷:

- ✗ 基础检测器限制模型整体能力;
- ✗ 在轻量模型上训练不稳定;

Efficient Semi-DETR



在线异构框架

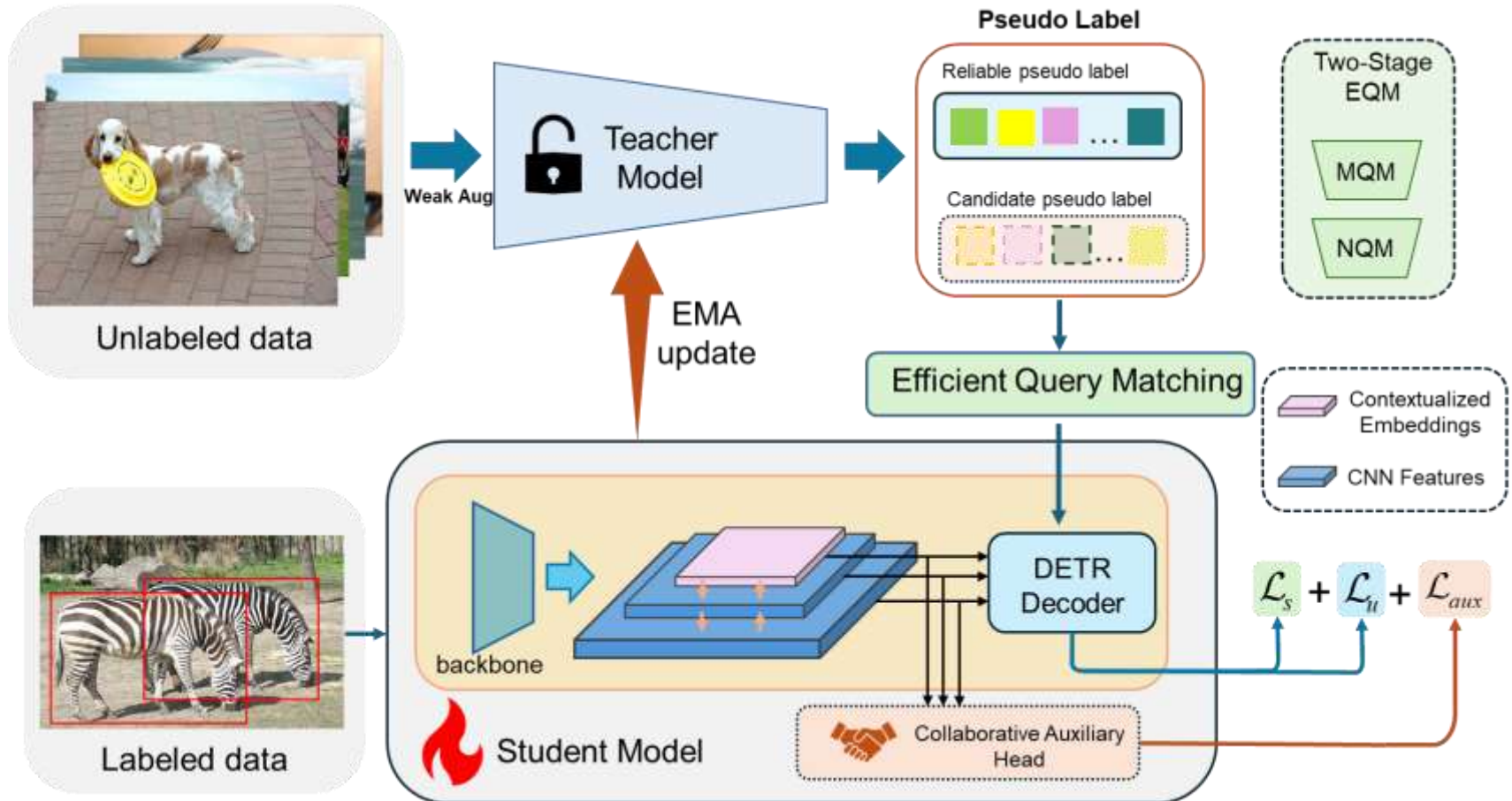
算法优势:

✓ **首个**在线异构框架的半监督目标检测算法;

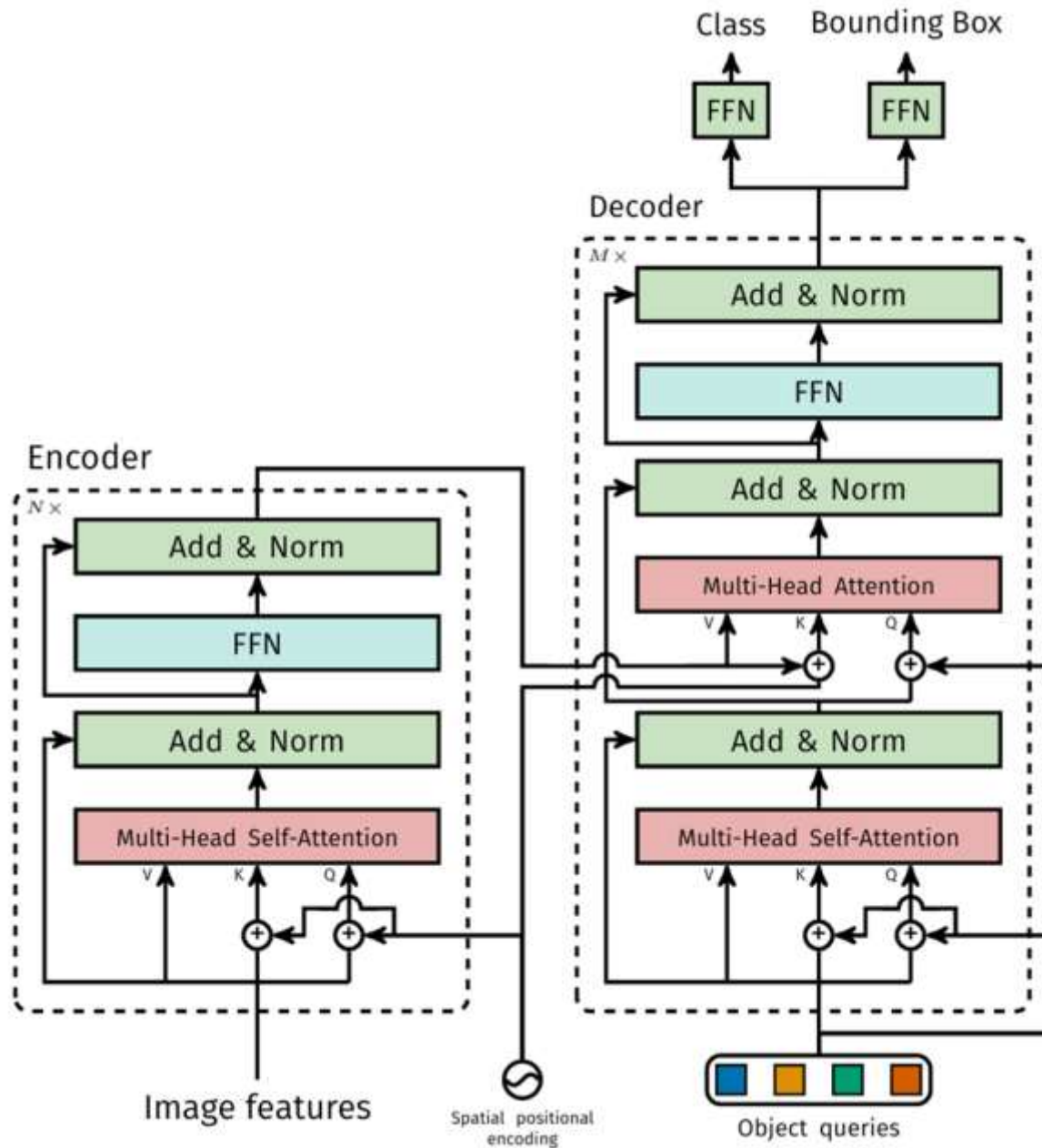
✓ **取长补短**: 解决了离线和同构框架的缺陷;

✓ **高效推理**;

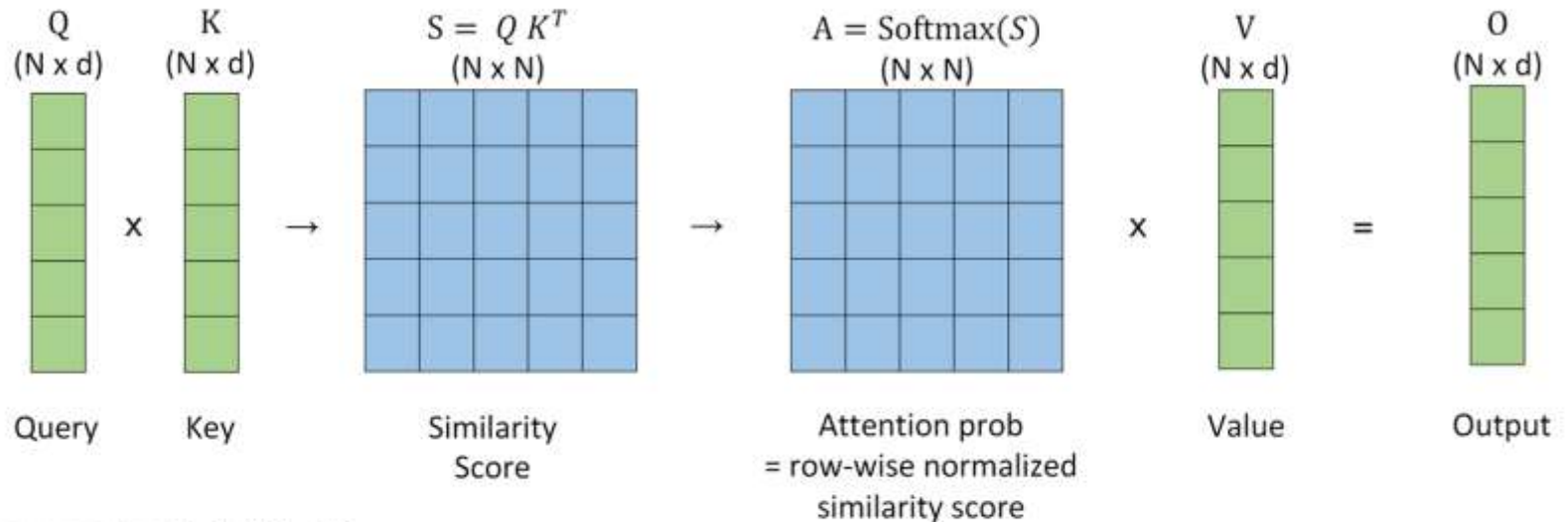
Efficient Semi-DETR



DETR



Transformer



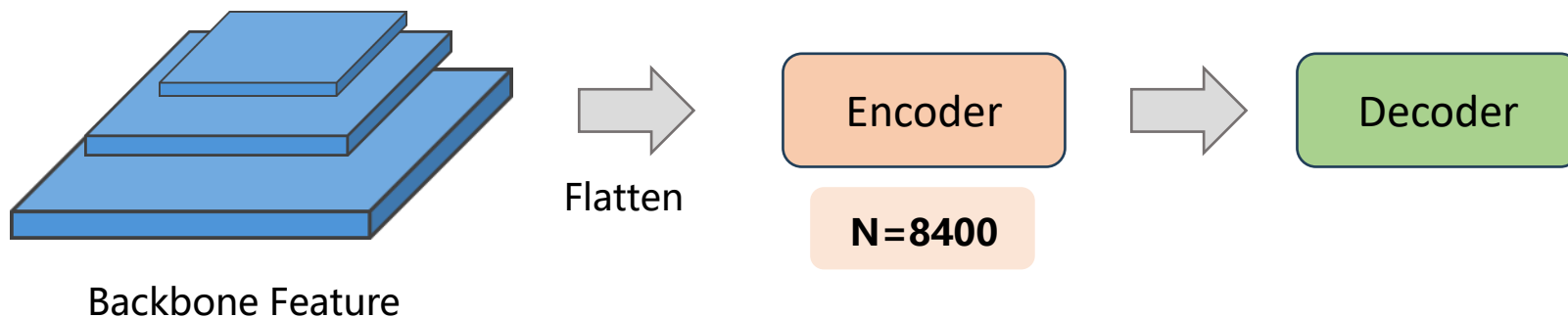
Typical sequence length N: 1K – 8K
Head dimension d: 64 – 128

$$\text{Softmax}([s_1, \dots, s_N]) = \left[\frac{e^{s_1}}{\sum_i e^{s_i}}, \dots, \frac{e^{s_N}}{\sum_i e^{s_i}} \right]$$

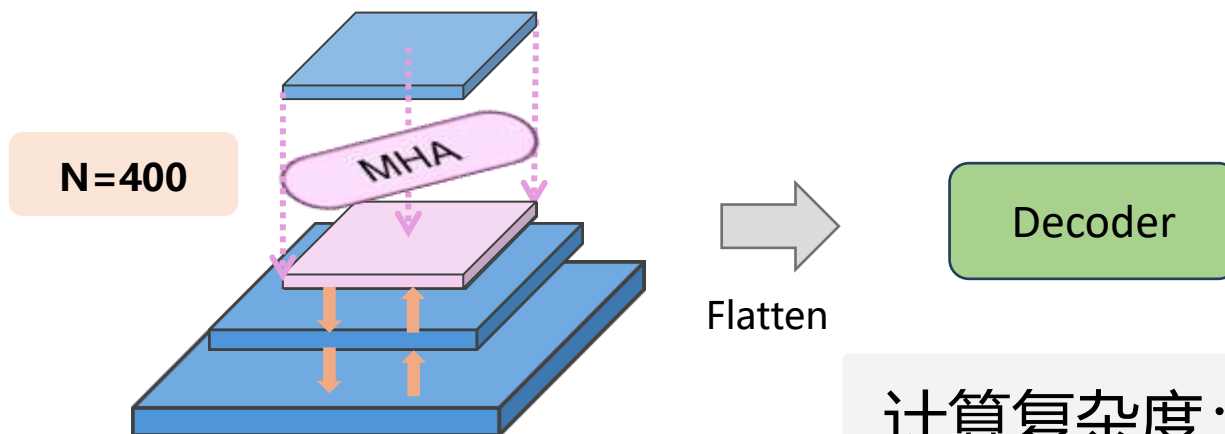
$$\mathbf{O} = \text{Softmax}(\mathbf{QK}^T)\mathbf{V}$$

- 1: Load \mathbf{Q} , \mathbf{K} by blocks from HBM, compute $\mathbf{S} = \mathbf{QK}^T$, write \mathbf{S} to HBM.
- 2: Read \mathbf{S} from HBM, compute $\mathbf{P} = \text{softmax}(\mathbf{S})$, write \mathbf{P} to HBM.
- 3: Load \mathbf{P} and \mathbf{V} by blocks from HBM, compute $\mathbf{O} = \mathbf{PV}$, write \mathbf{O} to HBM.
- 4: Return \mathbf{O} .

Efficient Semi-DETR

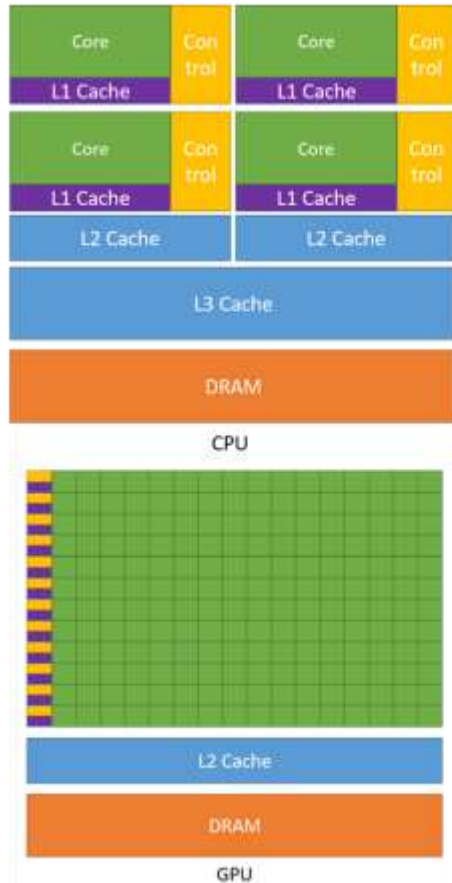


DETR流程图

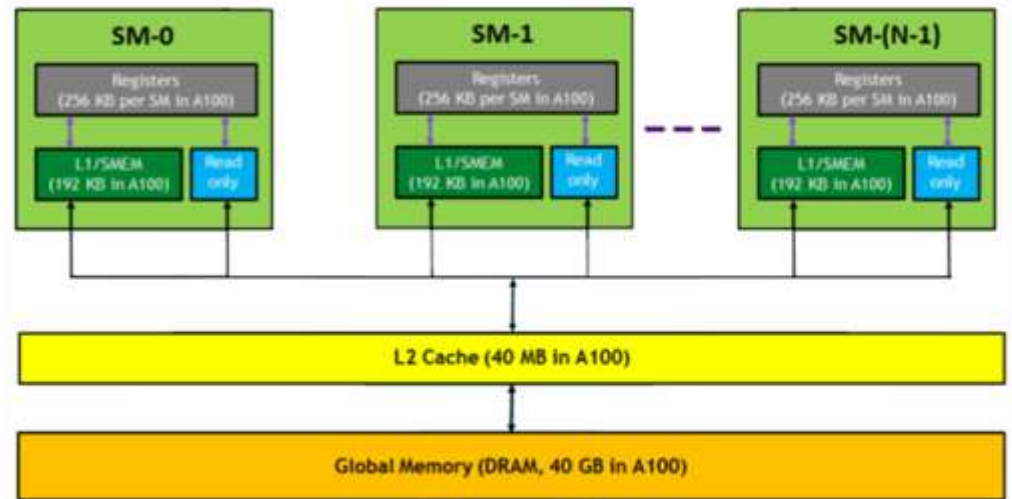


计算复杂度: $O(N^2)$

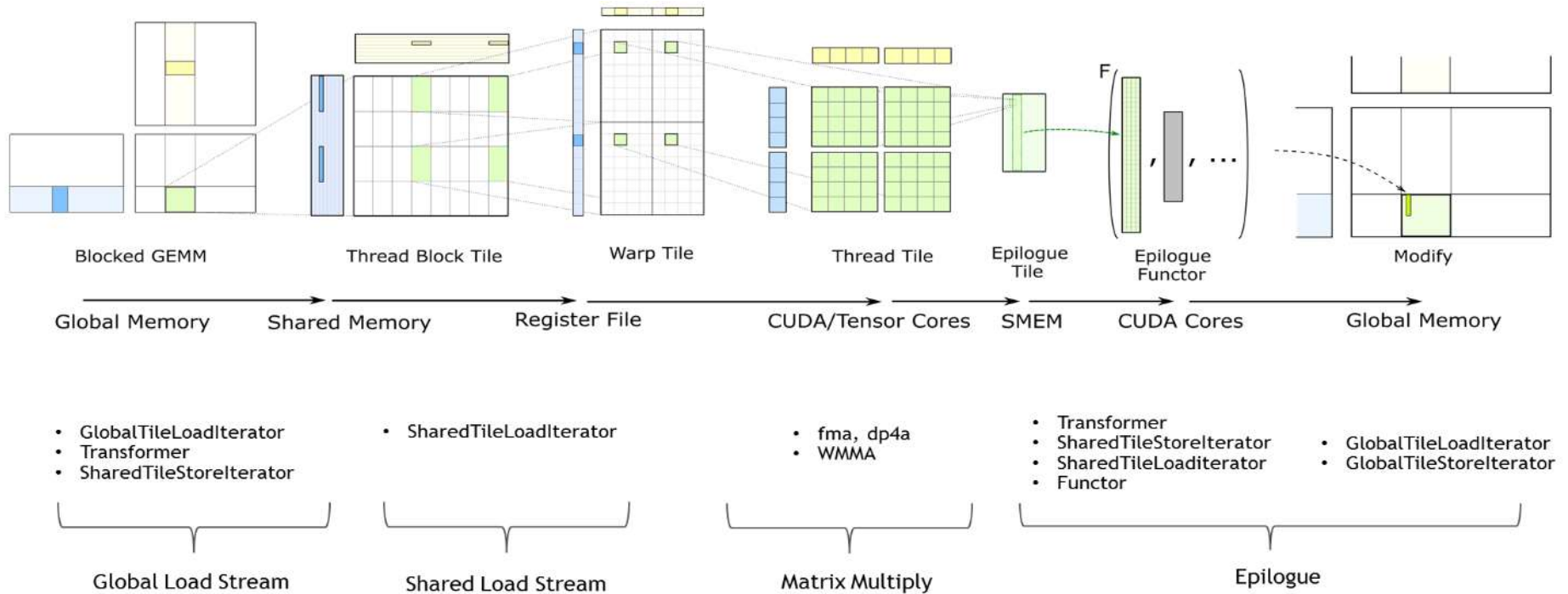
Efficient Semi-DETR



The GPU Devotes More Transistors to Data Processing

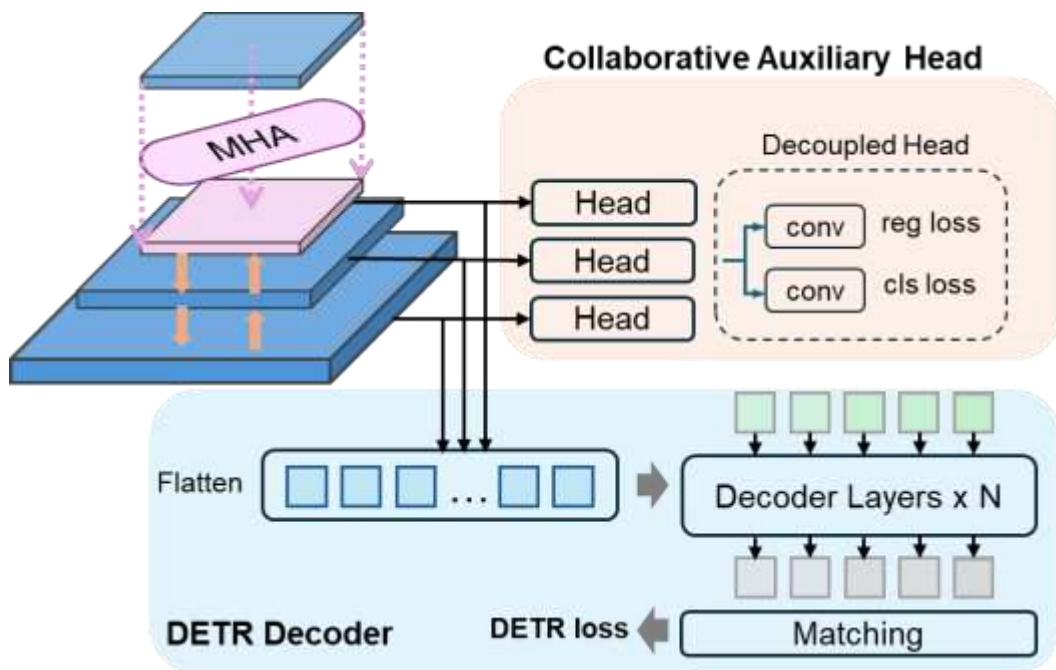


Efficient Semi-DETR



CUTLASS GEMM Structural Model

Efficient Semi-DETR



$$\mathcal{L}_u = \mathbb{I}(\hat{s} \geq \tau_{high}^c)(\mathcal{L}^{detr} + \mathcal{L}_{reg}^{aux}) + \mathbb{I}(\hat{s} > \tau_{low})\mathcal{L}_{cls}^{aux}$$

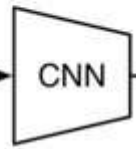
算法优势:

✓ 极大的降低了训练和推理开销;

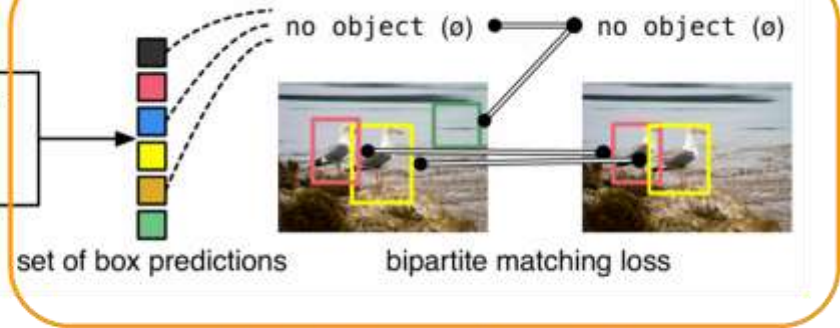
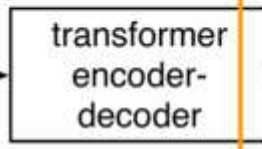
✓ 解决了训练初期监督信号稀疏导致模型收敛难的问题;

✓ 辅助分支不会增加推理开销;

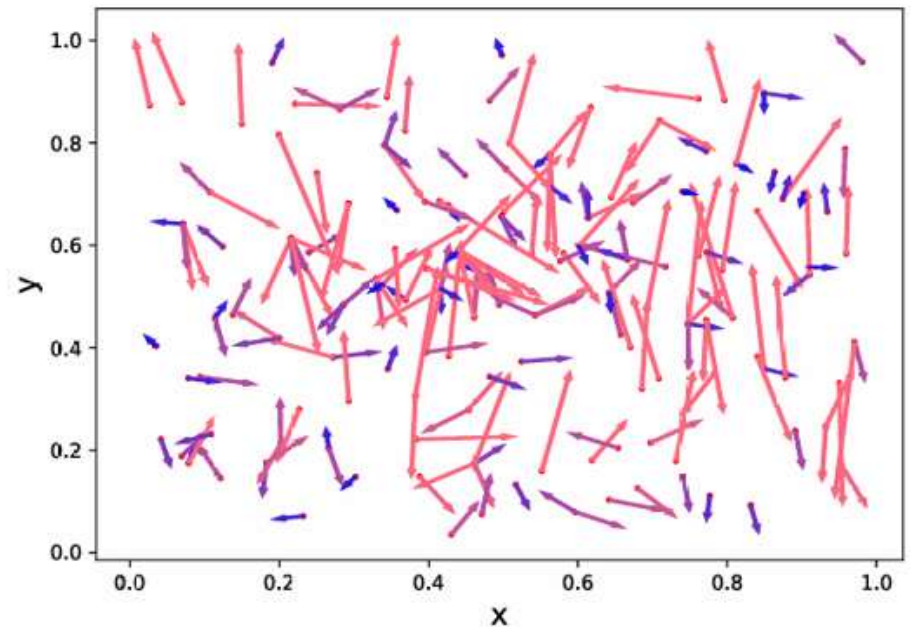
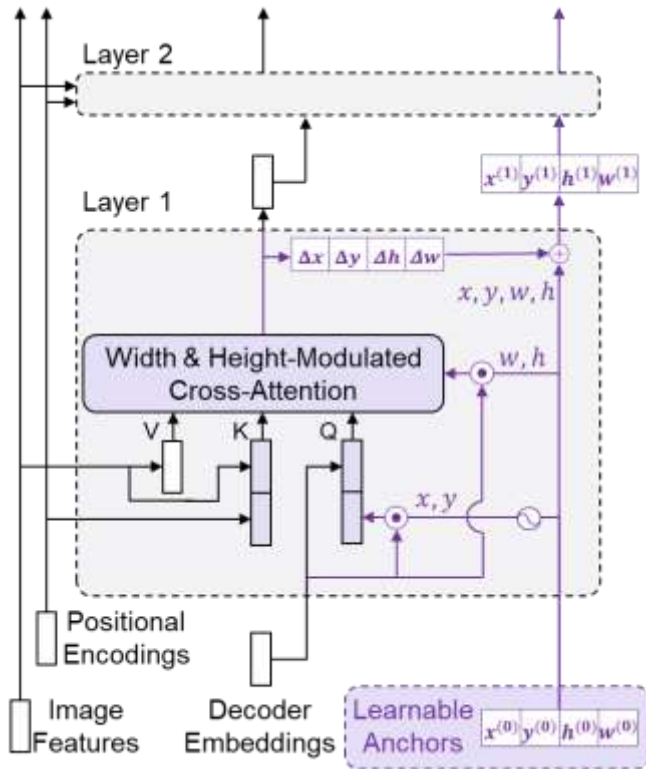
DETR-based



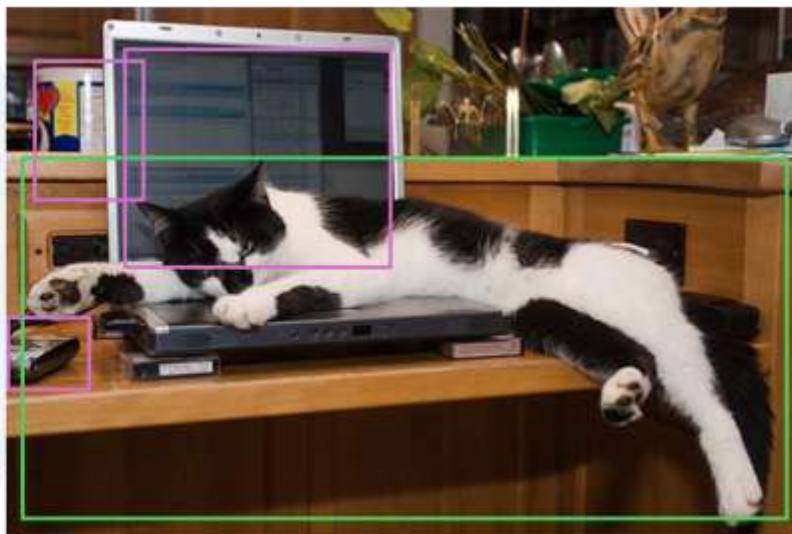
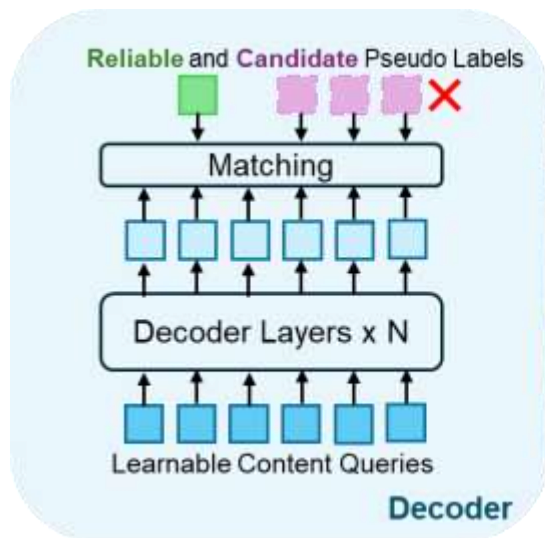
set of image features



bipartite graph matching

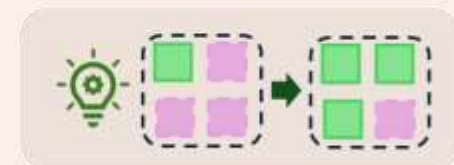


Efficient Semi-DETR



$$\mathcal{L}_{cls}(\bar{x}^u, \text{green square, purple squares}) \text{ Easy!}$$

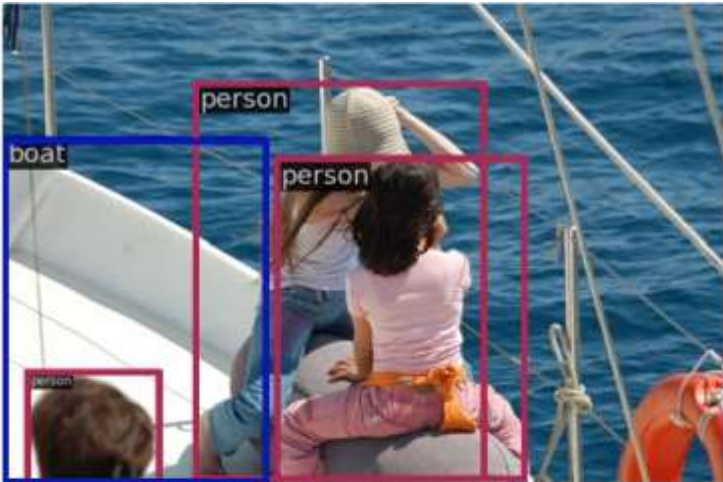
$$\mathcal{L}_{reg}(\bar{x}^u, \text{green square}) \text{ Hard!}$$



Collaborative Auxiliary Head

基于最优匹配的Decoder无法利用低质量伪标签中有效监督信息，
基于卷积的辅助头能够解耦分类和回归任务！

Efficient Semi-DETR



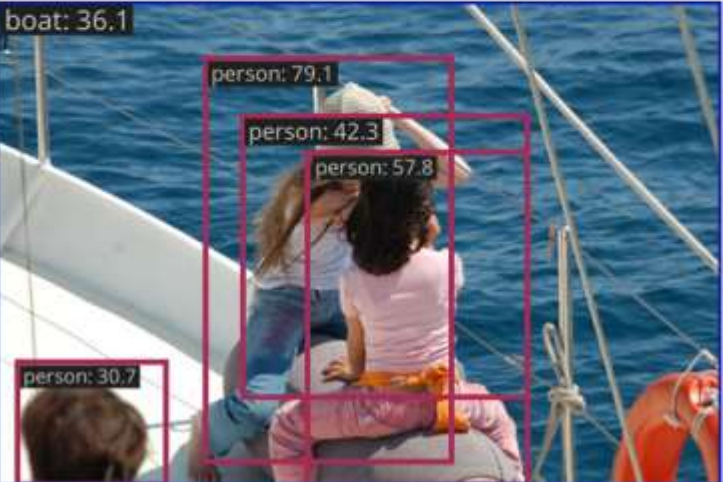
(a) Ground Truth



(b) Efficient PAN-style Encoder



(c) Vanilla PAN



(d) One-to-many Assignment

实验结果

	Method	1%	5%	10%	GFLOPs
Two-stage	Faster RCNN(supervised) [8]	10.02 \pm 0.38	20.82 \pm 0.13	26.44 \pm 0.11	202
	STAC [22]	13.97 \pm 0.35	24.38 \pm 0.12	28.64 \pm 0.21	202
	Instant Teaching [37]	18.05 \pm 0.15	26.75 \pm 0.05	30.40 \pm 0.05	202
	Humber Teacher [23]	16.96 \pm 0.38	27.70 \pm 0.15	31.61 \pm 0.28	-
	Unbiased Teacher [18]	18.33 \pm 0.19	26.65 \pm 0.31	29.56 \pm 0.24	204
	Soft Teacher [31]	20.46 \pm 0.39	30.74 \pm 0.08	34.04 \pm 0.14	202
	LabelMatch [2]	25.81 \pm 0.28	32.70 \pm 0.18	35.49 \pm 0.17	-
	PseCo [12]	22.43 \pm 0.36	32.50 \pm 0.08	36.06 \pm 0.21	-
One-stage	FCOS(supervised) [24]	9.05 \pm 0.31	20.69 \pm 0.22	26.01 \pm 0.15	-
	Unbiased Teacher V2 [19]	25.40 \pm 0.36	31.85 \pm 0.09	35.08 \pm 0.02	201
	DSL [3]	22.03 \pm 0.28	30.87 \pm 0.24	36.22 \pm 0.18	201
	ARSL [16]	25.36 \pm 0.32	34.41 \pm 0.16	38.50 \pm 0.05	-
	Efficient Teacher [30]	23.76 \pm 0.13	34.11 \pm 0.09	37.90 \pm 0.04	169
	Consistent Teacher [28]	25.3	36.1	40.0	208
DETR-based	DINO(supervised) [33]	18.00 \pm 0.21	29.50 \pm 0.16	35.00 \pm 0.12	279
	Omni-DETR [27]	27.60	37.70	41.30	-
	Semi-DETR [34]	30.50 \pm 0.30	40.10 \pm 0.15	43.50 \pm 0.10	>279
	Sparse Semi-DETR [21]	30.90 \pm 0.23	40.80 \pm 0.12	44.30 \pm 0.01	>279
	Efficient Semi-DETR	31.18 \pm 0.19	41.05 \pm 0.13	44.92 \pm 0.12	133

10% COCO数据集实验结果

实验：推理延迟

Method	$AP_{50:95}$	$FPS_{bs=1}$
Consistent Teacher	40.00	29.6
ARSL	38.50 ± 0.05	37.7
Efficient Teacher	37.90 ± 0.04	59.3
DINO(supervised)	35.00 ± 0.14	18.3
Omni-DETR	41.30	21.5
Semi-DETR	43.50 ± 0.10	18.1
Efficient Semi-DETR	44.92 ± 0.12	98.3

训练成本比较：

Semi-DETR: 大约需要280G显存，在八卡A100上需要20+小时；

Ours: 大约需要50G显存，在八卡RTX 3090需要约12小时。