High-Performance Visual Tracking Algorithms

Dong Wang

wdice@dlut.edu.cn

Dalian University of Technology

Visual Object Tracking

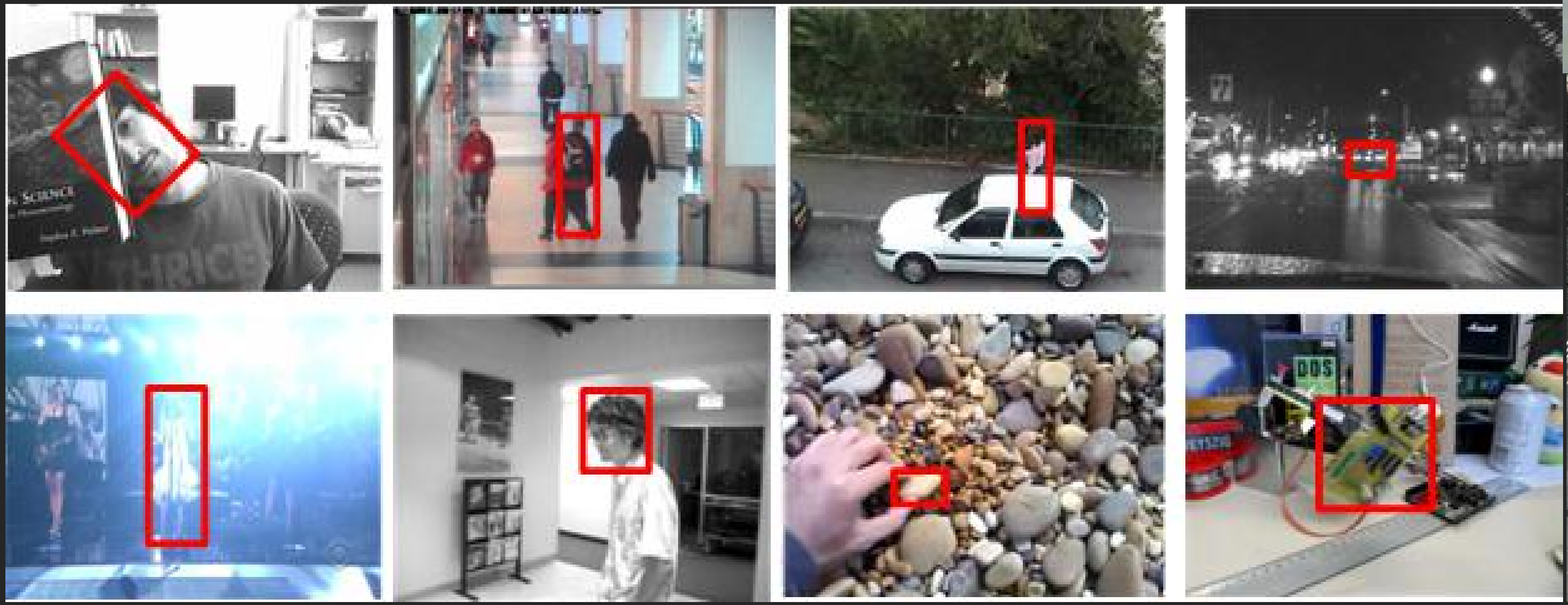
- **Goal**

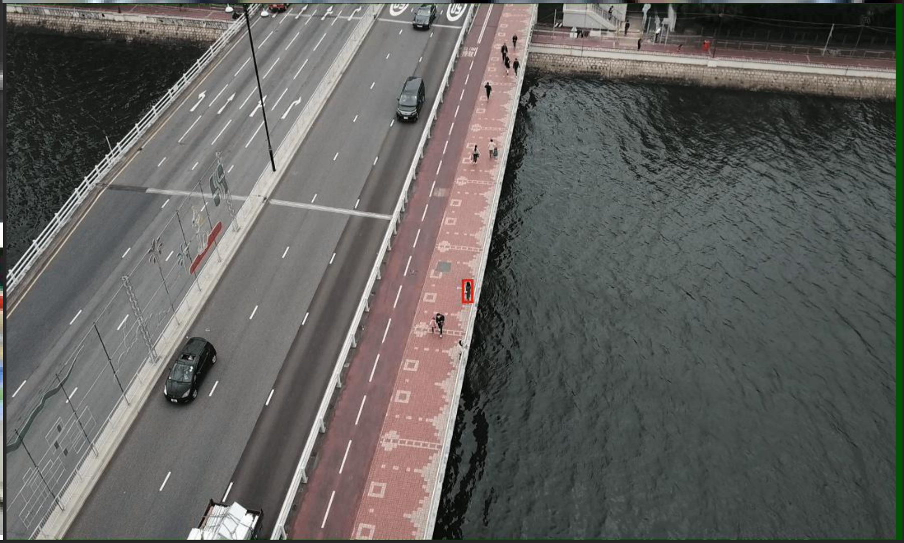
  Track an arbitrary object in a video given its initial location

  Single-object, **Model-free**

- **Challenges**

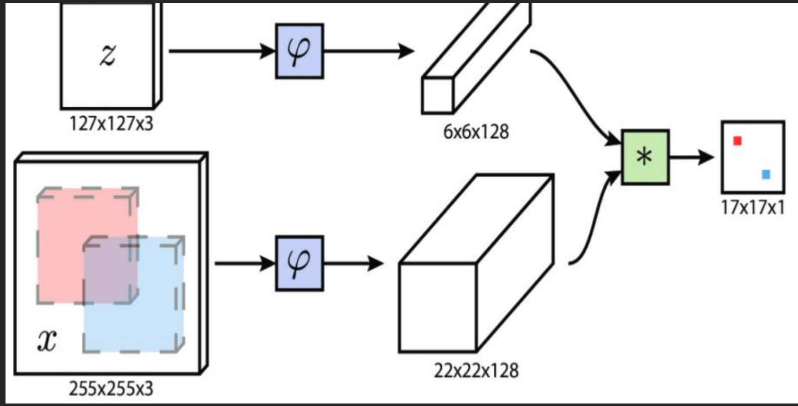
  Occlusion, Light Change, Background Clutter, etc.



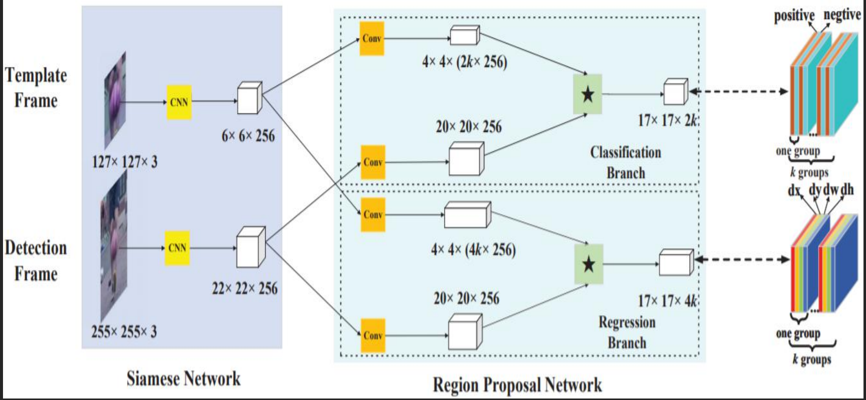


Visual Object Tracking: One-shot vs Online

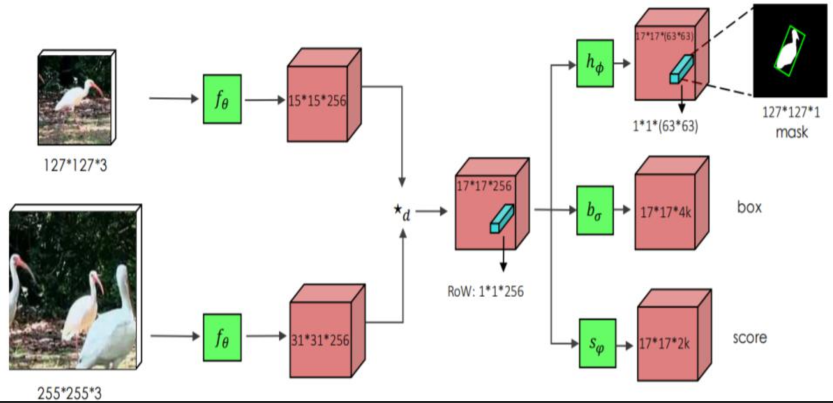
SiamFC (ECCVW16)



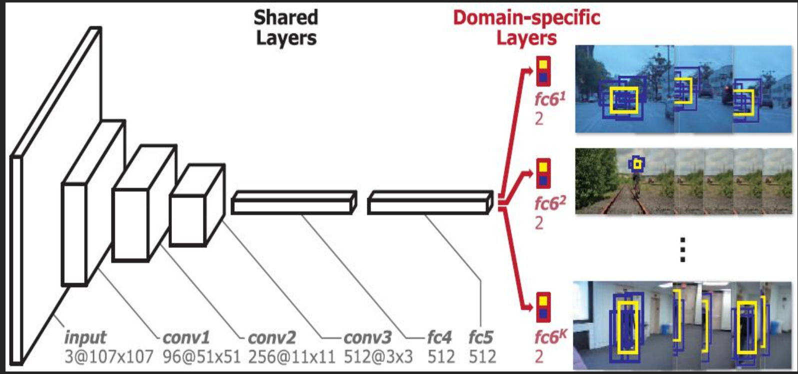
SiamRPN (CVPR18)



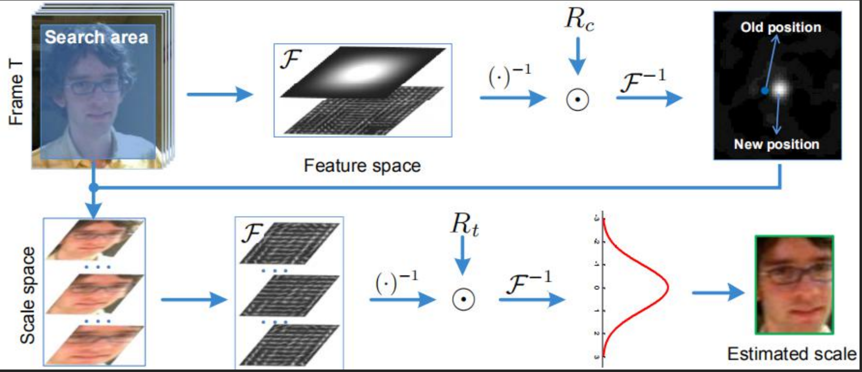
SiamMask (CVPR19)



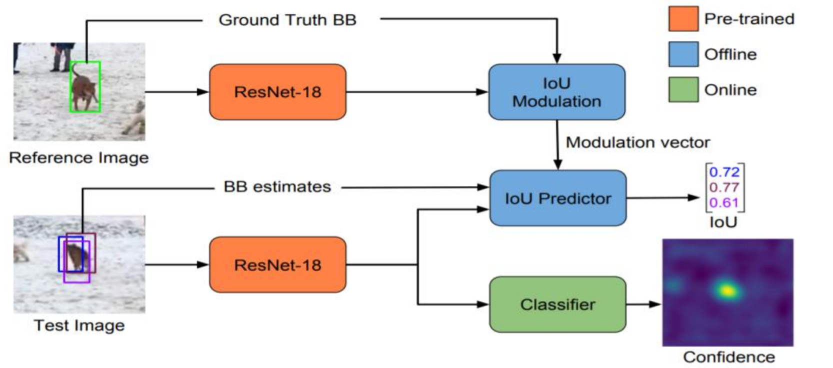
MDNet (CVPR16)



ECO (ICCV17)

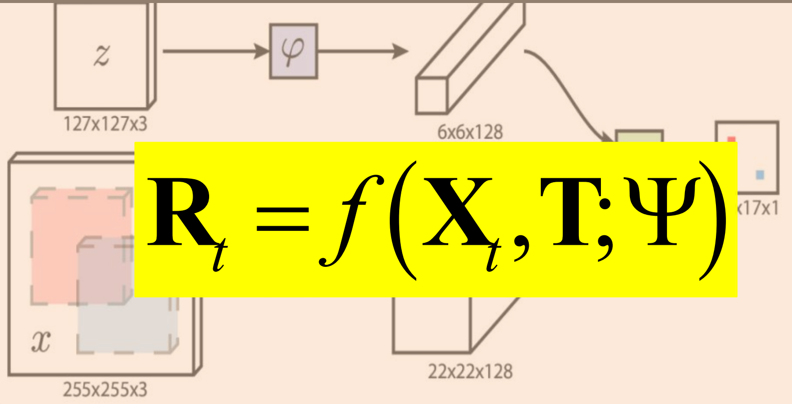


ATOM (CVPR19)



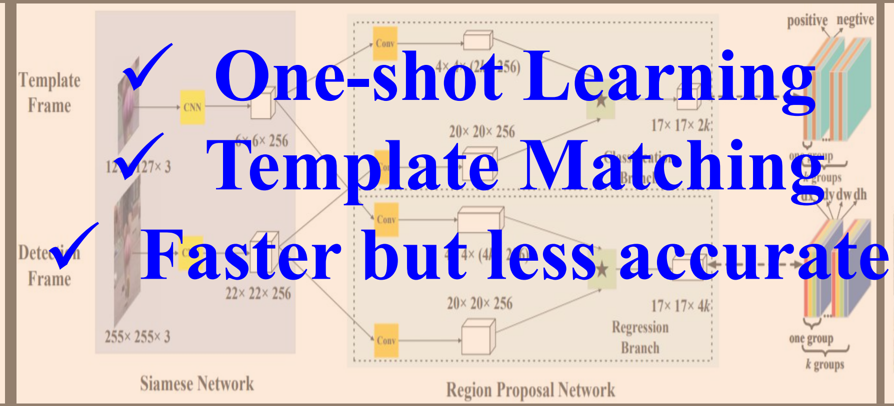
Visual Object Tracking: One-shot vs Online

SiamFC (ECCVW16)



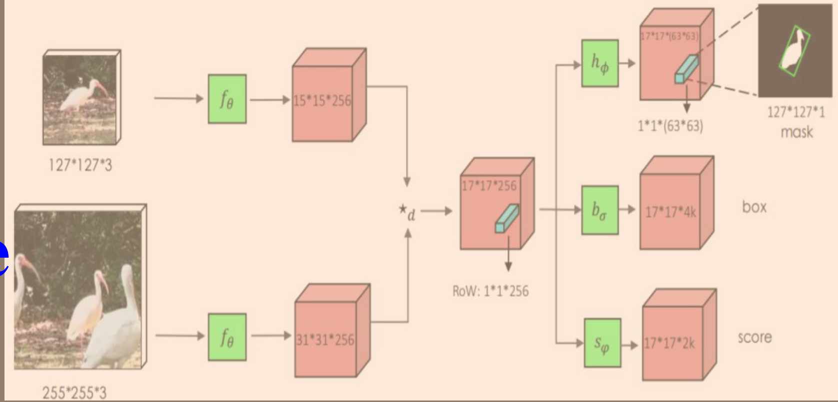
$$\mathbf{R}_t = f\left(\mathbf{X}_t, \mathbf{T}; \Psi\right)$$

SiamRPN (CVPR18)

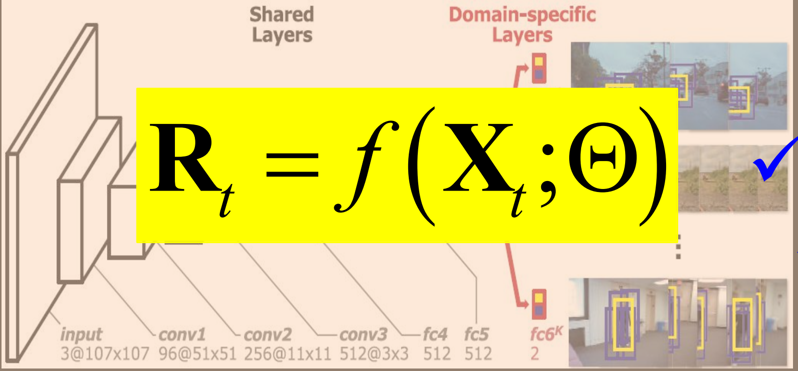


- ✓ One-shot Learning
- ✓ Template Matching
- ✓ Faster but less accurate

SiamMask (CVPR19)

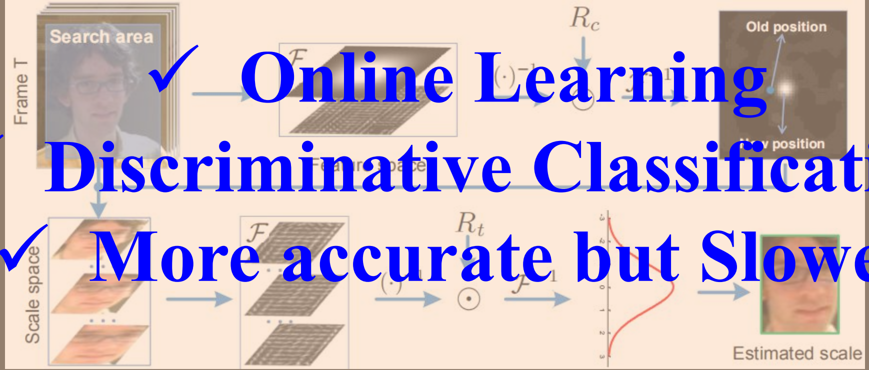


MDNet (CVPR16)



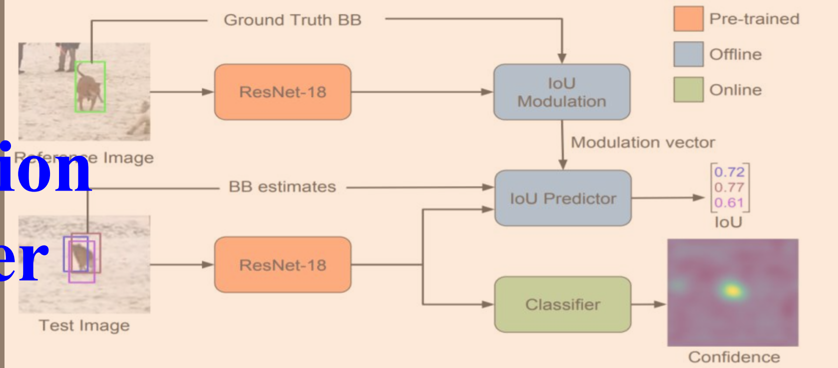
$$\mathbf{R}_t = f\left(\mathbf{X}_t; \Theta\right)$$

ECO (ICCV17)



- ✓ Online Learning
- ✓ Discriminative Classification
- ✓ More accurate but Slower

ATOM (CVPR19)

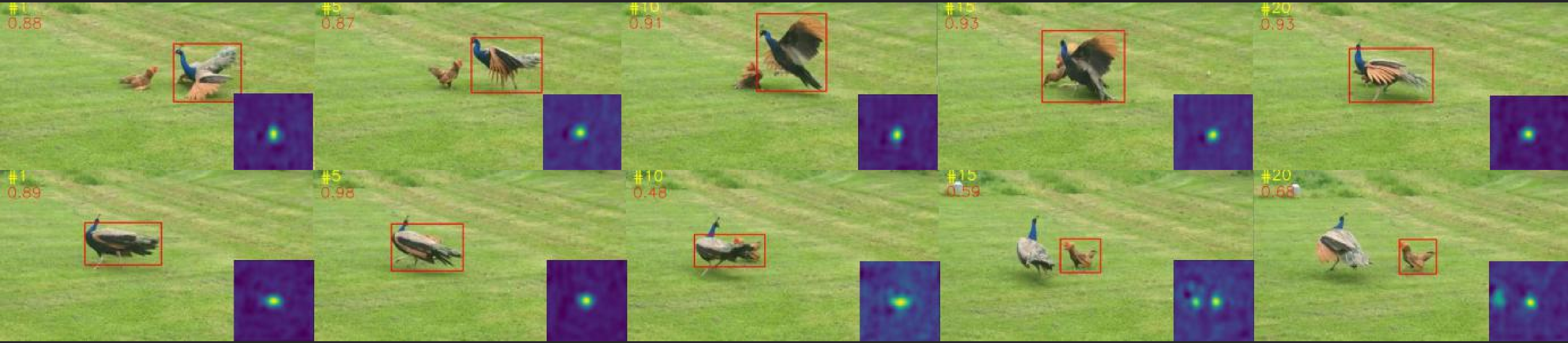


High-Performance Visual Tracking Algorithms

- **Strong Robustness** —Noisy Update Handling→ **Meta-Update** (CVPR2020)
- **High-Accuracy** —Transformer→ **TransT,** (CVPR2021) **Stark** (ICCV2021)
- **Plug-and-Play** —General Scale Estimation→ **Alpha-Refine** (CVPR2021)
- **Moblie Setting** —One-shot NAS in Siamese→ **LightTrack** (CVPR2021)

Tracking with Meta-Updater

High-Performance Long-Term Tracking with Meta-Updater (LTMU)



Kenan Dai, Yunhua Zhang, **Dong Wang**, Jianhua Li, Huchuan Lu, Xiaoyun Yang. High-Performance Long-Term Tracking with Meta-Updater, CVPR, 2020.

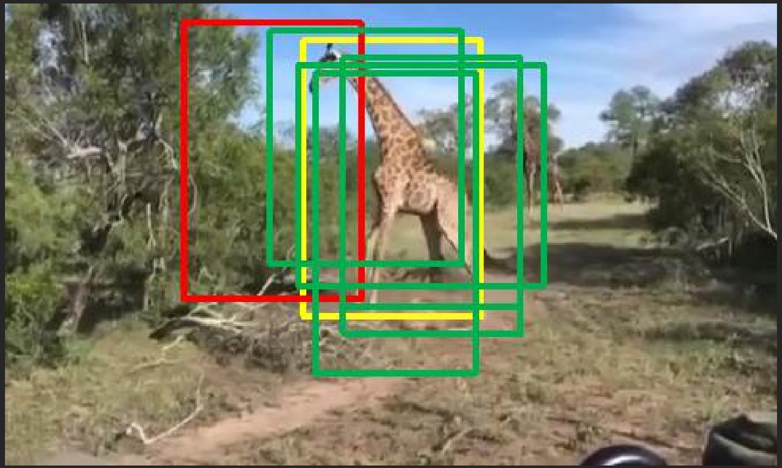
➢ Code: https://github.com/Daikenan/LTMU

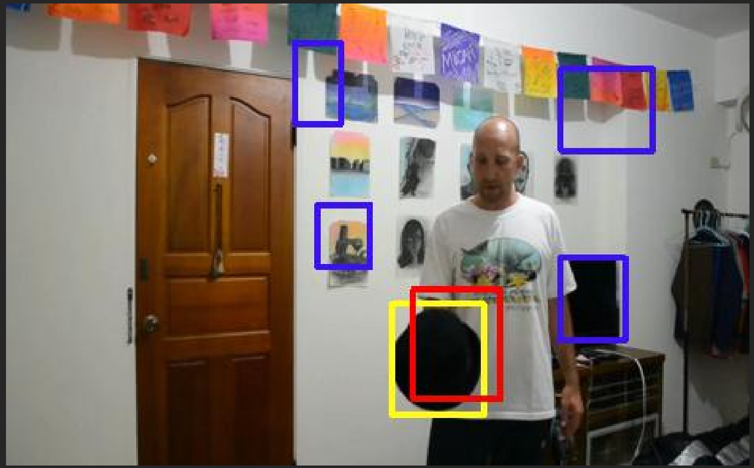
Model Update in Visual Object Tracking

MDNet(CVPR16), RT-MDNet(ECCV18),...

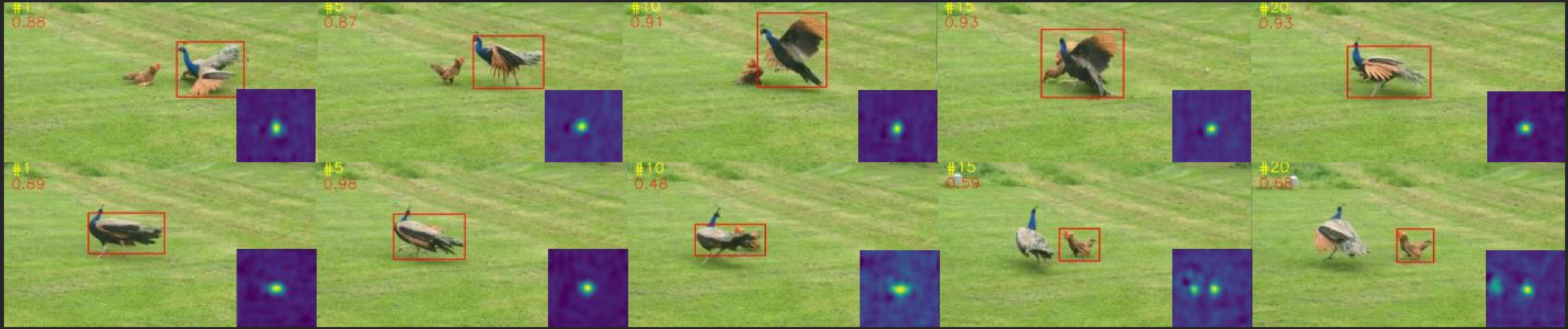
$$\mathbf{R}_t = f\left(\mathbf{X}_t;\boxed{\Theta}\right)$$

Sparse Sampling





Dense Sampling

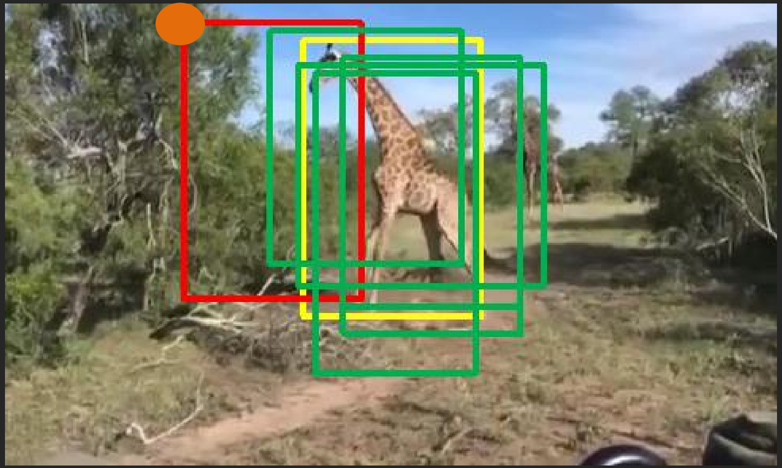


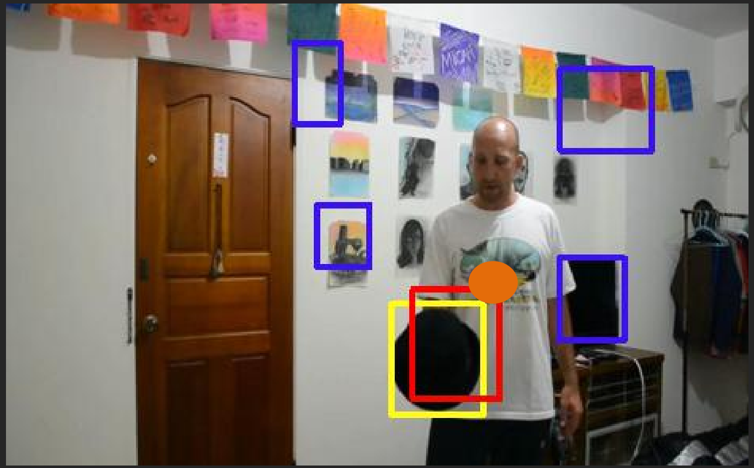
ECO(ICCV17), ATOM(CVPR19), DiMP(ICCV19),...

Model Update in Visual Object Tracking

MDNet(CVPR16), RT-MDNet(ECCV18),...

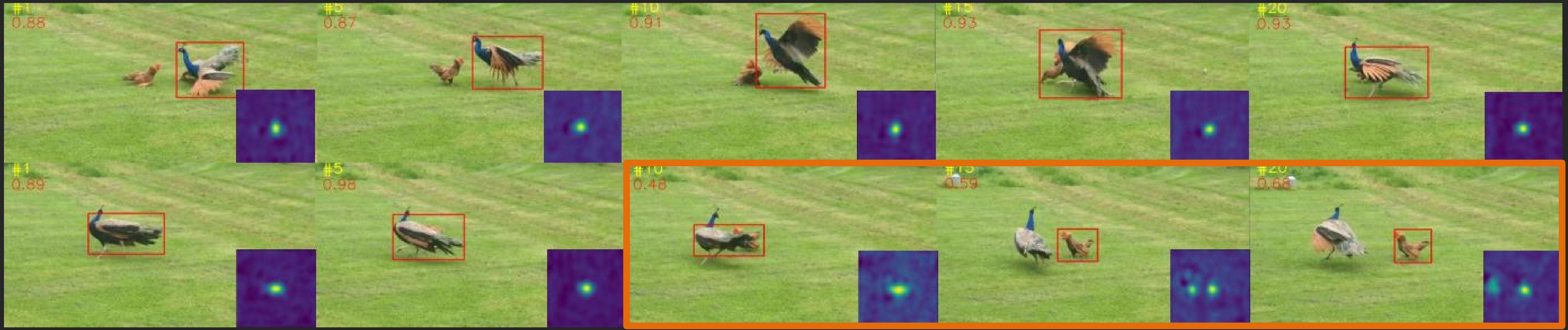
$$\mathbf{R}_t = f\left(\mathbf{X}_t; \boxed{\Theta}\right)$$





Nosiy Observations

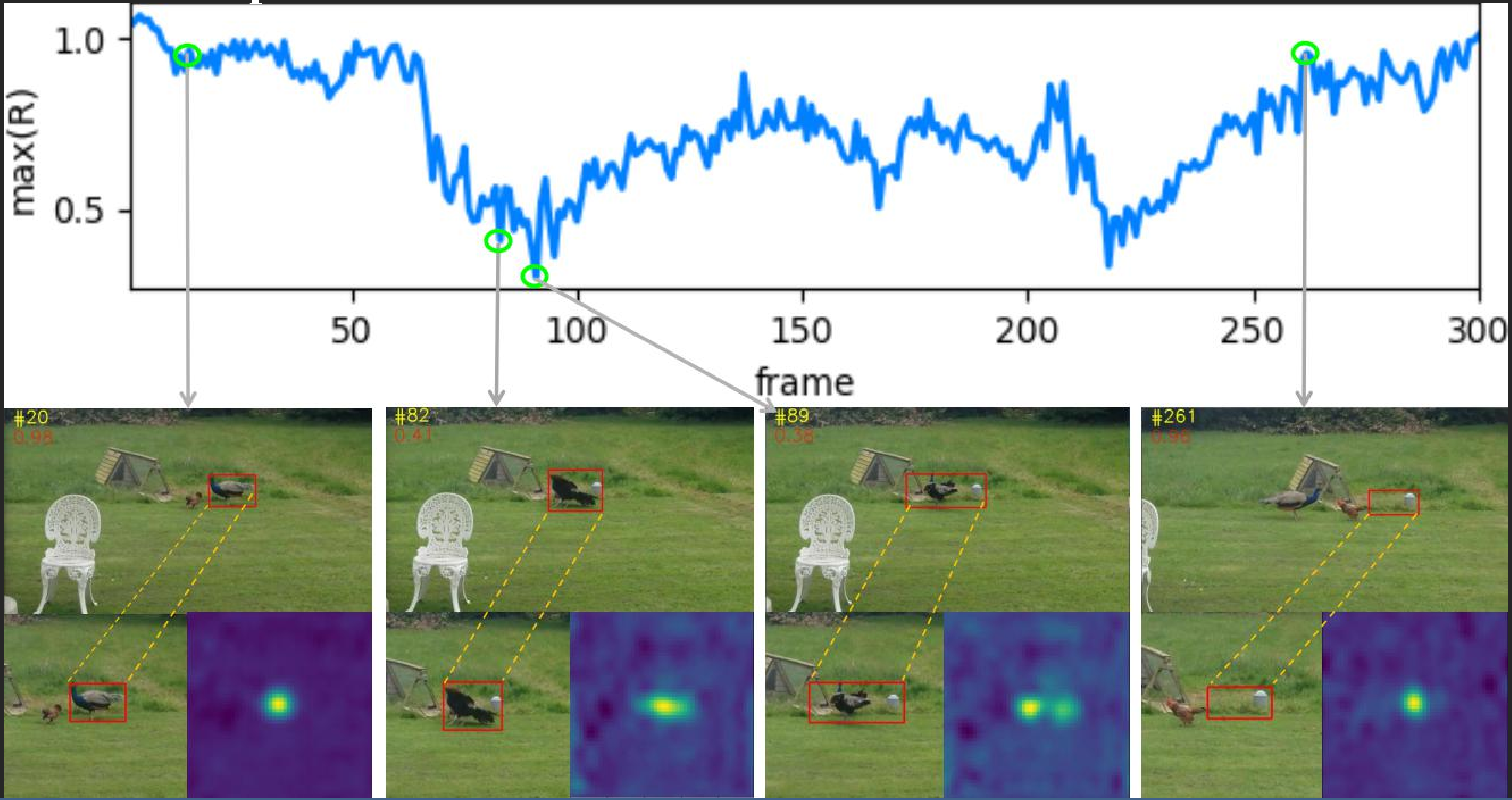
! ! ! ! !



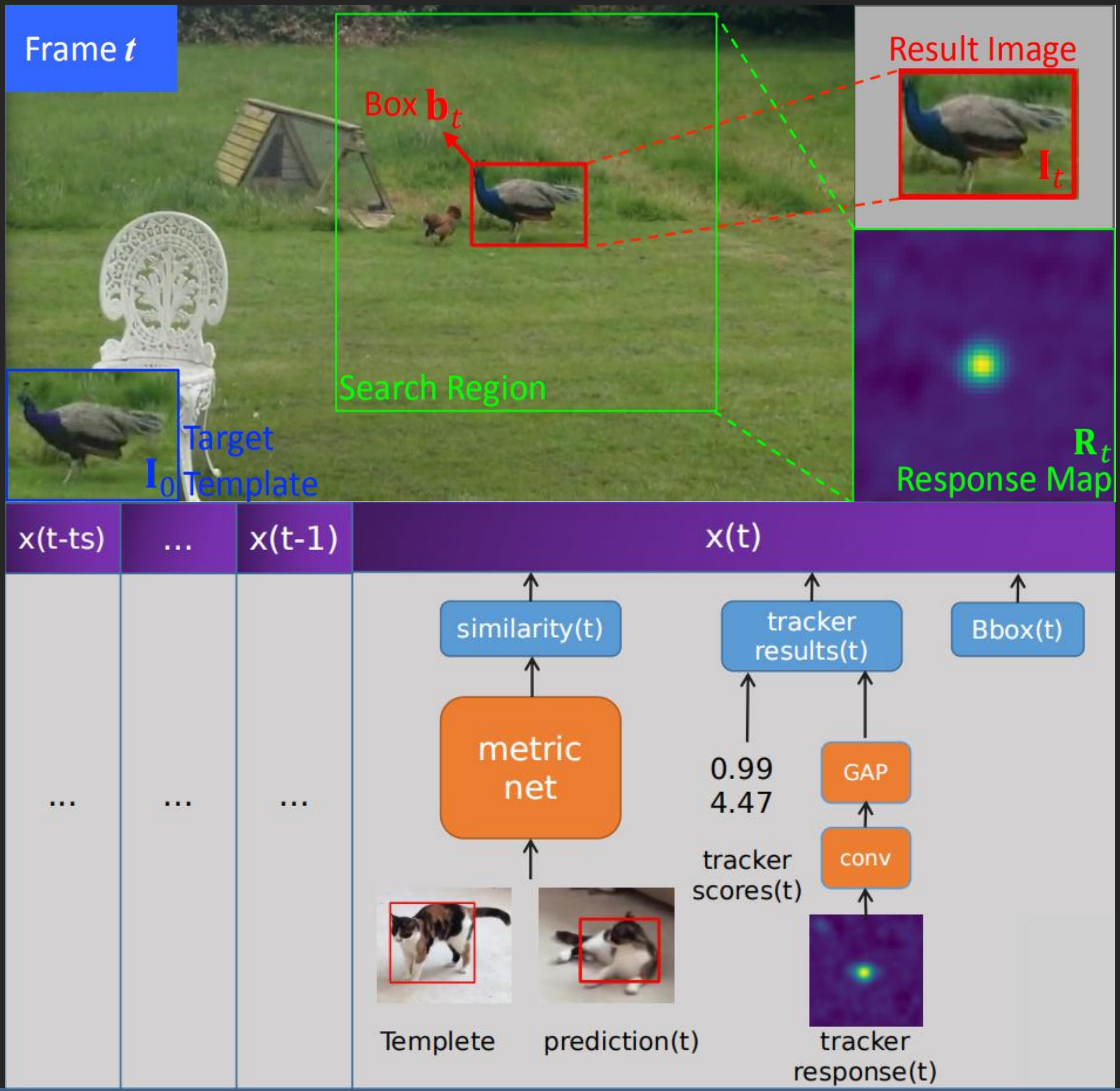
ECO(ICCV17), ATOM(CVPR19), DiMP(ICCV19), ...

Tracking with Meta-Updater

- **Max Response?**



Tracking with Meta-Updater



- **Sequential Information**
  - **Geometric Cue:**

$$\mathbf{b}_t = [x_t, y_t, w_t, h_t]$$

  - **Discriminative Cue:**
    - ✓ **Score:**

$$s_t^C = \max(\mathbf{R}_t)$$

    - ✓ **Map:**

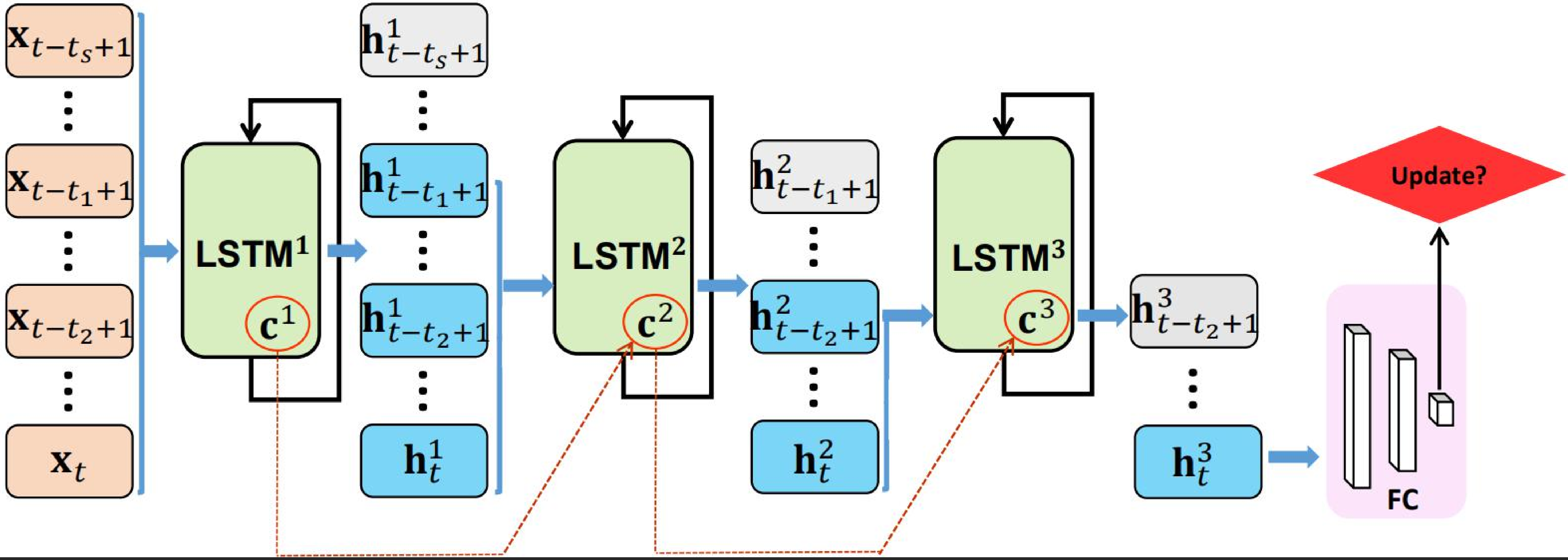
$$\mathbf{v}_t^R = f^R(\mathbf{R}_t; \mathbf{W}^R)$$

  - **Appearance Cue:**

$$s_t^A = \left\| f^A(\mathbf{I}_t, \mathbf{W}^A) - f^A(\mathbf{I}_0, \mathbf{W}^A) \right\|_2$$

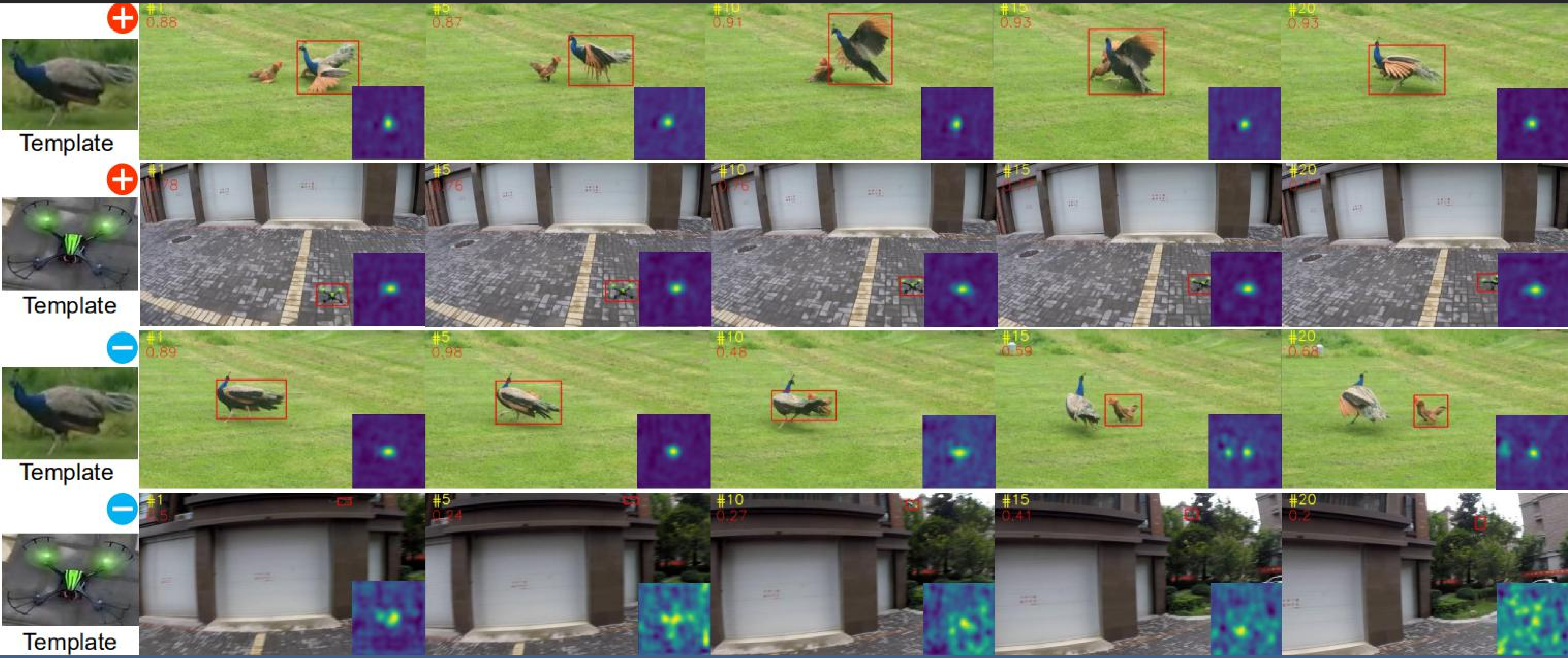
Tracking with Meta-Updater

➢ **Three-stage Cascaded LSTMs**



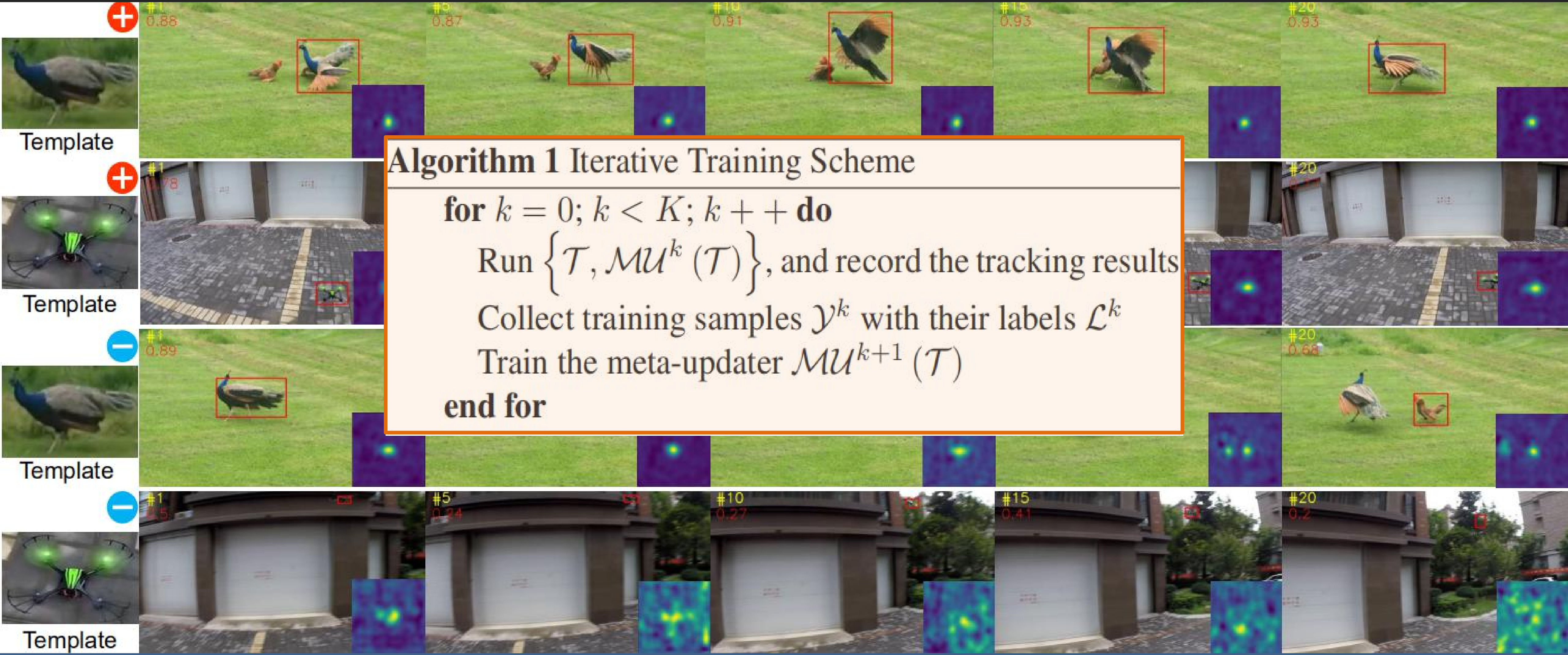
Tracking with Meta-Updater

➢ Training



Tracking with Meta-Updater

➢ Training



Algorithm 1 Iterative Training Scheme

for $k = 0; k < K; k++$ **do**

  Run $\left\{\mathcal{T}, \mathcal{MU}^{k}\left(\mathcal{T}\right)\right\}$, and record the tracking results

  Collect training samples $\mathcal{Y}^{k}$ with their labels $\mathcal{L}^{k}$

  Train the meta-updater $\mathcal{MU}^{k+1}\left(\mathcal{T}\right)$

end for

Long-Term Tracking with Meta-Updater

➢ Generalization Ability

+MU = Basetracker+Meta-Updater

+LTMU = Basetracker+Meta-Updater+Global Detection+Verifier+Bbox refine

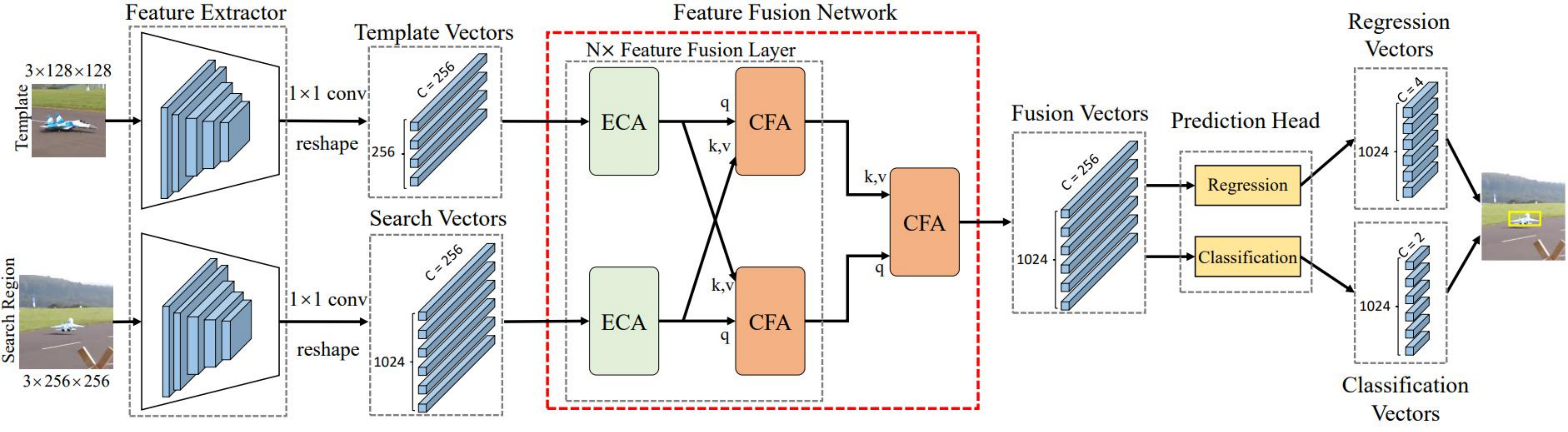
Tracker	LaSOT(AUC)	VOT2020 LT(F)	VOT2018 LT(F)	TLP(AUC)
RT-MDNet	0.335	0.338	0.367	0.276
RT-MDNet+MU	0.354	0.396	0.407	0.337
ATOM	0.511	0.497	0.510	0.399
ATOM+MU	0.541	0.620	0.628	0.473
DiMP	0.568	0.573	0.587	0.514
DiMP+MU	0.594	0.641	0.649	0.564
DiMP+LTMU	0.602	0.691	-	0.572
PrDiMP	0.612	0.632	0.631	0.535
PrDiMP+MU	0.615	0.661	0.675	0.582
SuperDiMP	0.646	0.647	0.667	0.552
SuperDiMP+MU	0.658	0.704	0.707	0.595

Project

知乎中文讲解

Transformer Tracking

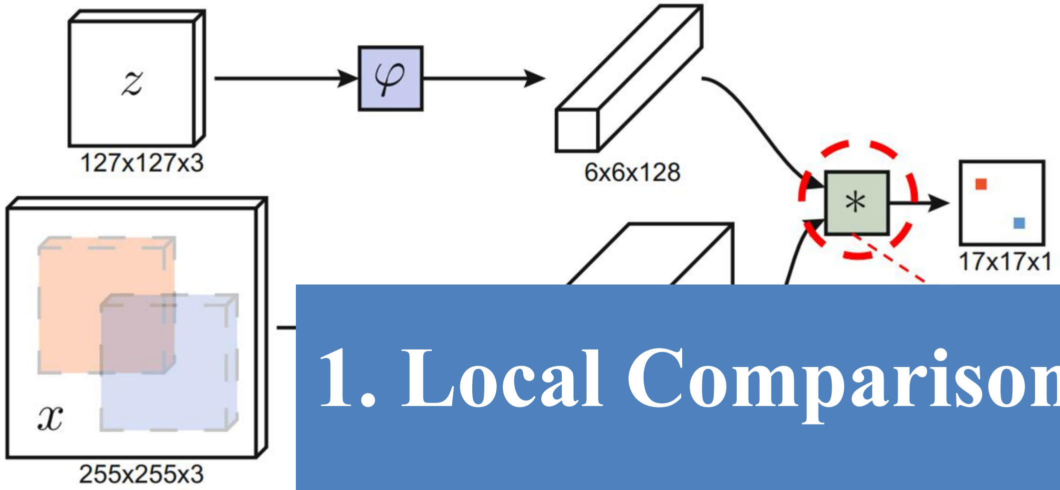
Transformer Tracking (TransT)

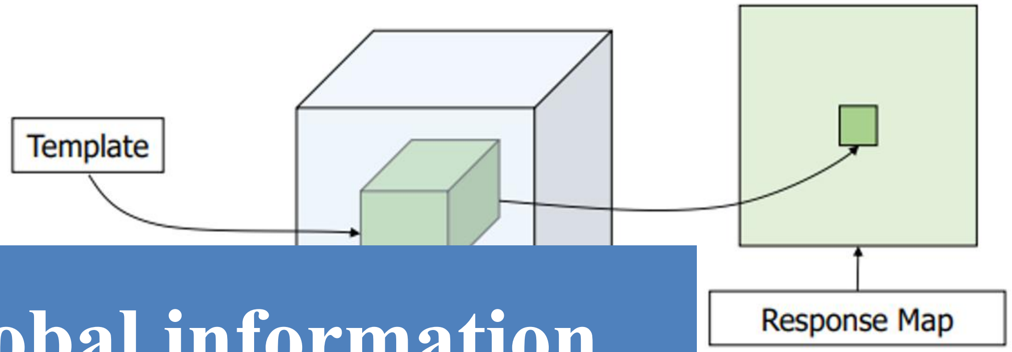


Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun yang, Huchuan Lu. Transformer Tracking. CVPR, 2021.

- Code: https://github.com/chenxin-dlut/TransT

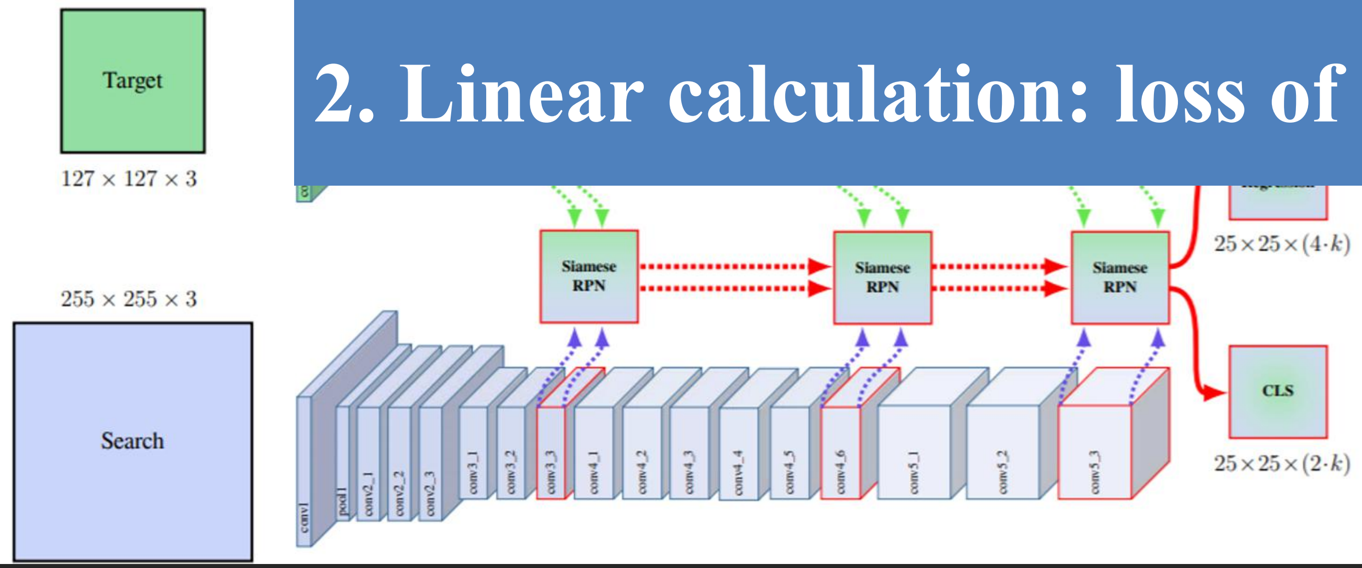
Transformer Tracking

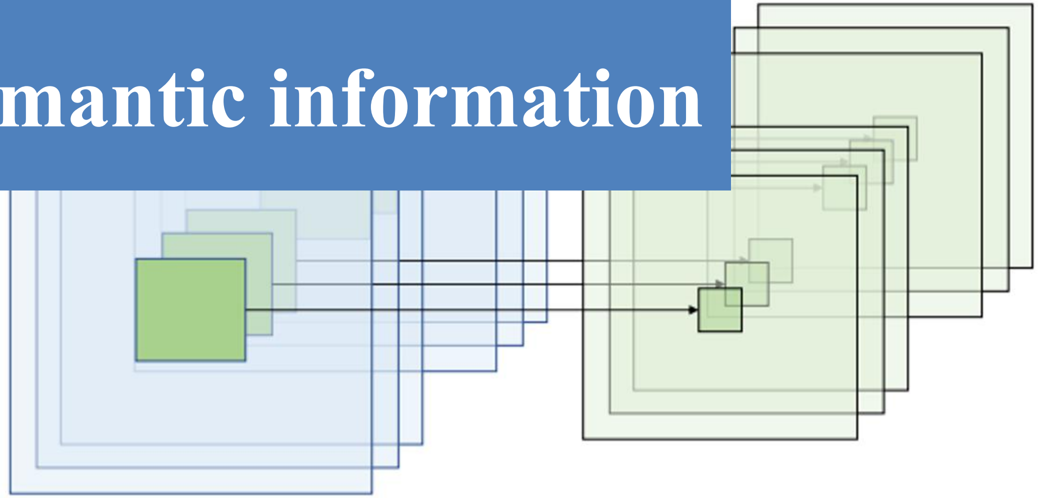




1. Local Comparison: lack of global information

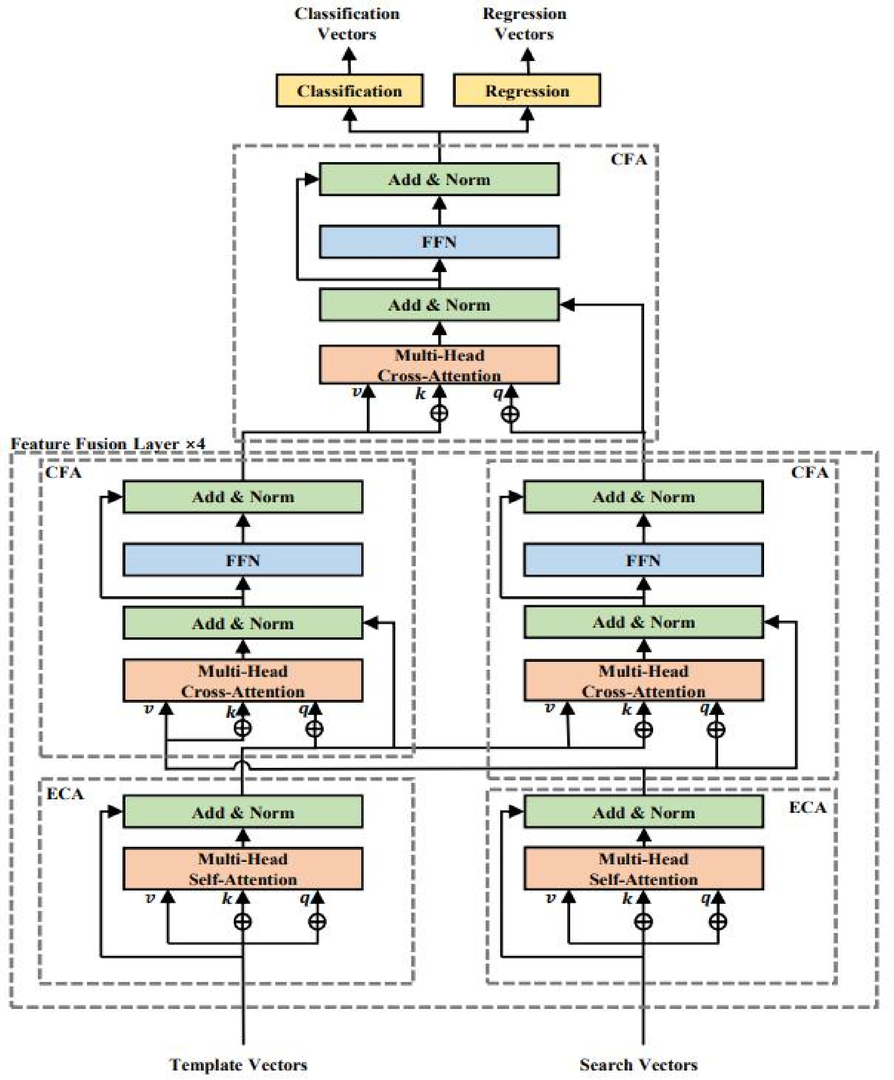
2. Linear calculation: loss of semantic information





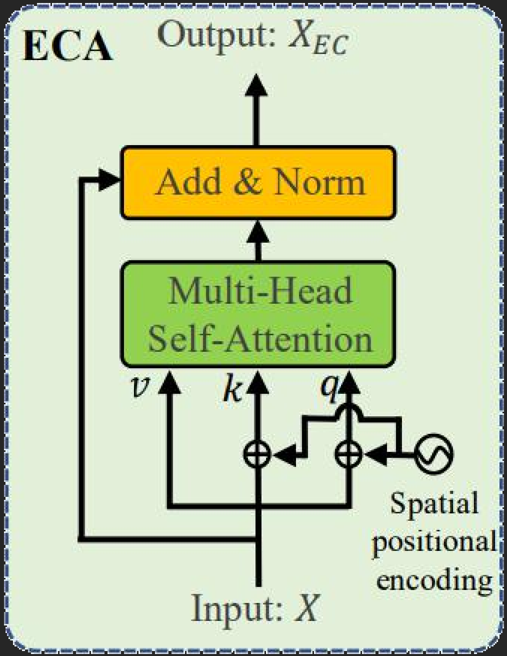
Transformer Tracking

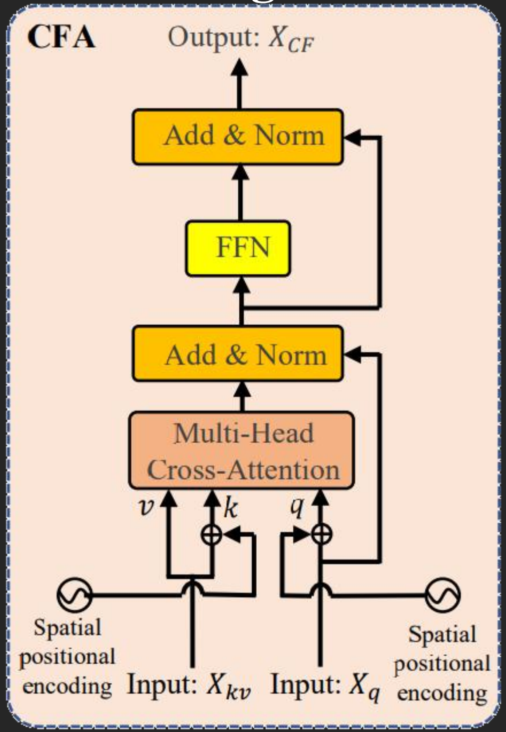
Our Feature Fusion Network



Cross-Feature Augment Module

Ego-Context Augment Module





- ✓ ECA based on self-attention and CFA based on cross-attention
- ✓ CFA performs feature fusion, retaining rich semantic information
- ✓ ECA and CFA establish dependence between long distance features and aggregate global information

Transformer Tracking

https://github.com/chenxin-dlut/TransT

DUT, IIAU-LAB

Transformer Tracking

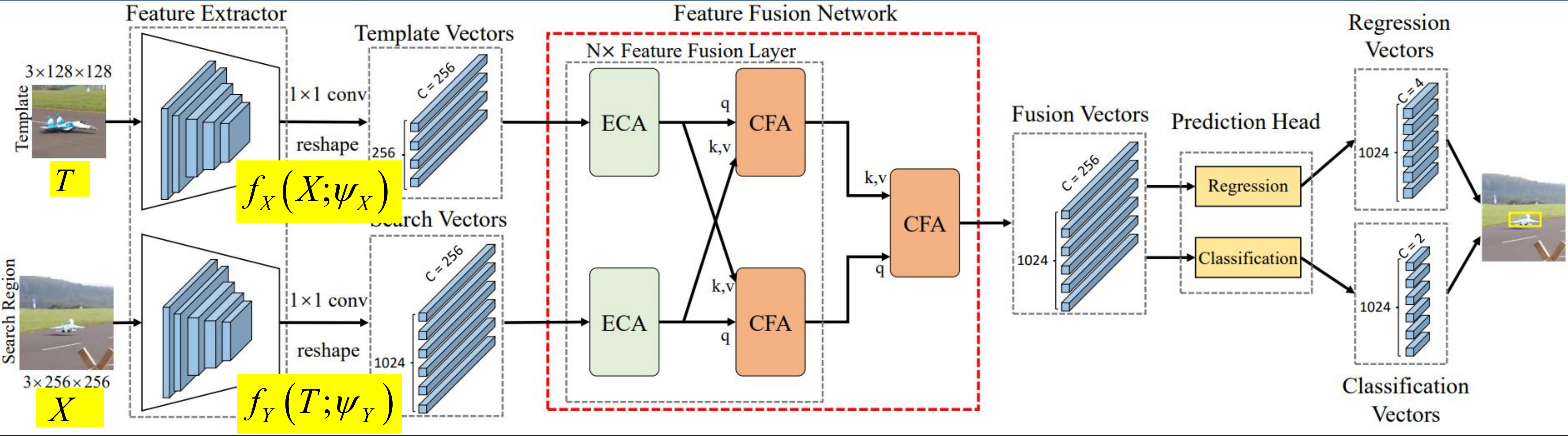
Large-scale Benchmark Results

Method	Source	LaSOT [14]			TrackingNet [30]			GOT-10k [19]		
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
TransT	Ours	**64.9**	**73.8**	**69.0**	**81.4**	**86.7**	**80.3**	**72.3**	**82.4**	**68.2**
TransT-N2	Ours	64.2	**73.5**	68.2	80.9	**86.4**	79.2	**69.9**	**80.1**	**65.9**
TransT-GOT	Ours	-	-	-	-	-	-	67.1	76.8	60.9
SiamR-CNN [39]	CVPR2020	**64.8**	72.2	-	**81.2**	85.4	**80.0**	64.9	72.8	59.7
Ocean [48]	ECCV2020	56.0	65.1	56.6	-	-	-	61.1	72.1	47.3
KYS [3]	ECCV2020	55.4	63.3	-	74.0	80.0	68.8	63.6	75.1	51.5
DCFST [49]	ECCV2020	-	-	-	75.2	80.9	70.0	63.8	75.3	49.8
SiamFC++ [44]	AAAI2020	54.4	62.3	54.7	75.4	80.0	70.5	59.5	69.5	47.9
PrDiMP [10]	CVPR2020	59.8	68.8	60.8	75.8	81.6	70.4	63.4	73.8	54.3
CGACD [13]	CVPR2020	51.8	62.6	-	71.1	80.0	69.3	-	-	-
SiamAttn [46]	CVPR2020	56.0	64.8	-	75.2	81.7	-	-	-	-
MAML [40]	CVPR2020	52.3	-	-	75.7	82.2	72.5	-	-	-
D3S [26]	CVPR2020	-	-	-	72.8	76.8	66.4	59.7	67.6	46.2
SiamCAR [16]	CVPR2020	50.7	60.0	51.0	-	-	-	56.9	67.0	41.5
SiamBAN [5]	CVPR2020	51.4	59.8	52.1	-	-	-	-	-	-
DiMP [2]	ICCV2019	56.9	65.0	56.7	74.0	80.1	68.7	61.1	71.7	49.2
SiamPRN++ [21]	CVPR2019	49.6	56.9	49.1	73.3	80.0	69.4	51.7	61.6	32.5
ATOM [9]	CVPR2019	51.5	57.6	50.5	70.3	77.1	64.8	55.6	63.4	40.2
ECO [8]	ICCV2017	32.4	33.8	30.1	55.4	61.8	49.2	31.6	30.9	11.1
MDNet [31]	CVPR2016	39.7	46.0	37.3	60.6	70.5	56.5	29.9	30.3	9.9
SiamFC [1]	ECCVW2016	33.6	42.0	33.9	57.1	66.3	53.3	34.8	35.3	9.8

46*fps*

66*fps*

Transformer Tracking

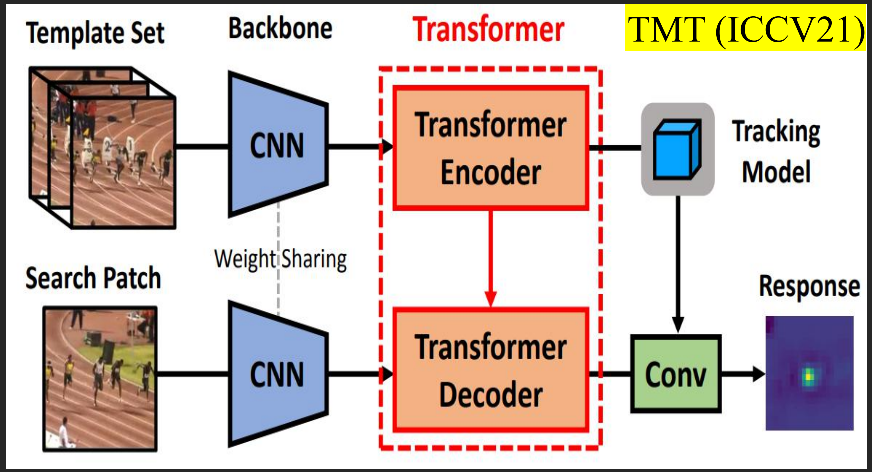


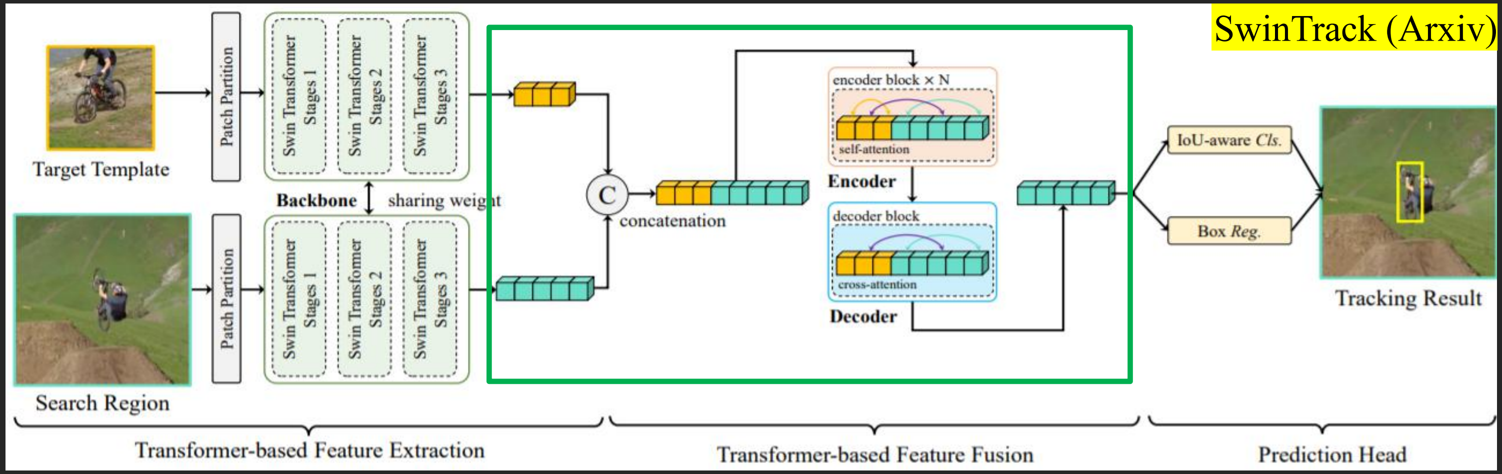
$$f(X,T;\psi)$$

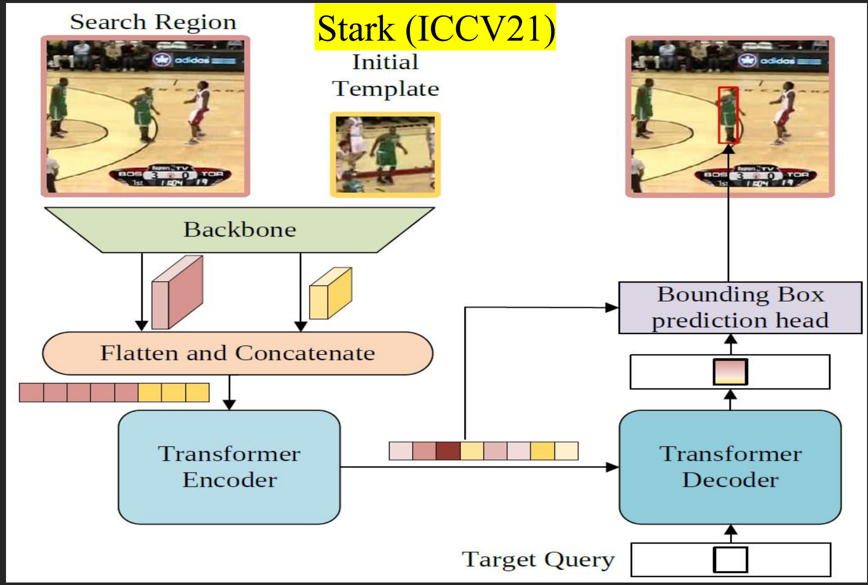
$$\Rightarrow f_X(X;\psi_X) * f_Y(T;\psi_Y)$$

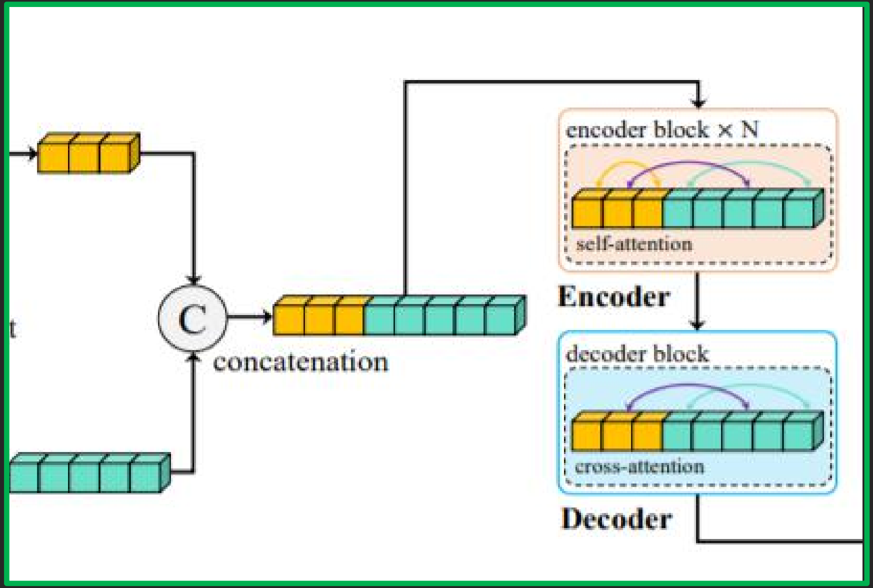
$$\Rightarrow f_F(f_X(X;\psi_X), f_Y(T;\psi_Y), \psi_F)$$

Transformer Tracking





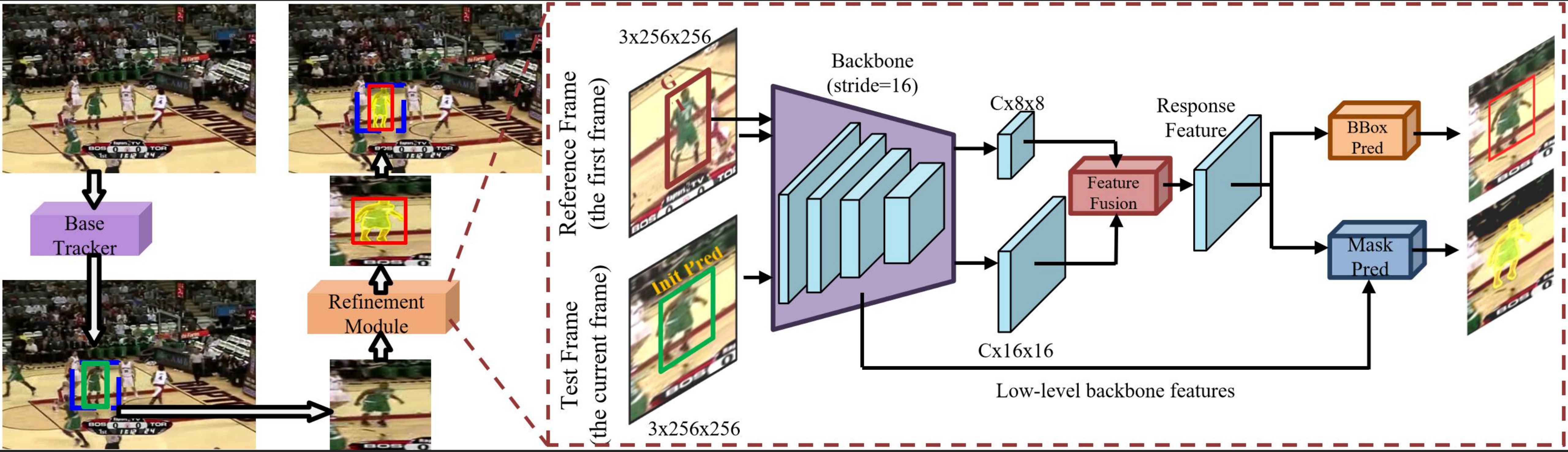




Tracker	SUC (%)	PRE (%)	NPRE (%)
SiamPRN++ [25]	49.6	-	56.9
DiMP [2]	56.9	53.4	65.0
Ocean [46]	56.0	56.6	65.1
SiamR-CNN [40]	64.8	-	72.2
TrSiam [41]	62.4	60.0	-
TrDiMP [41]	63.9	61.4	-
STMTrack [18]	60.6	63.3	69.3
TransT [7]	64.9	69.0	73.8
STARK-ST50 [44]	66.4	-	-
STARK-ST101 [44]	67.1	-	77.0
KeepTrack [32]	67.1	70.2	77.2
SwinTrack-T	66.7	70.6	75.8
SwinTrack-B	69.6	74.1	**78.6**
SwinTrack-B-384	**70.2**	**75.3**	78.4

Alpha-Refine

Alpha-Refine (AR)



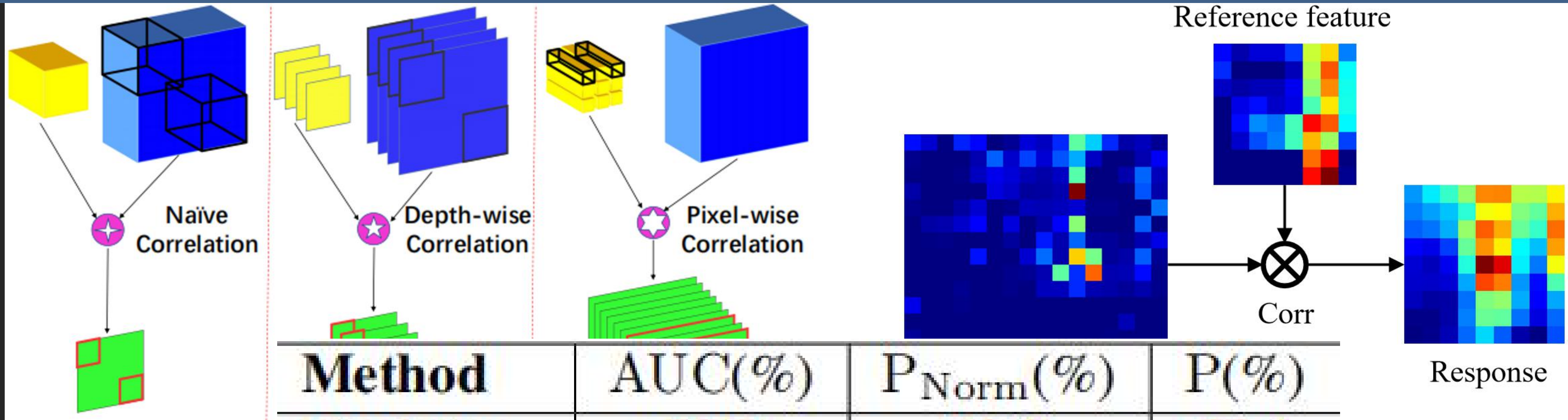
Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, Xiaoyun Yang. Alpha-Refine: Boosting Tracking Performance by Precise Bounding Box Estimation. CVPR, 2021.

➢ Code: https://github.com/MasterBin-IIAU/AlphaRefine

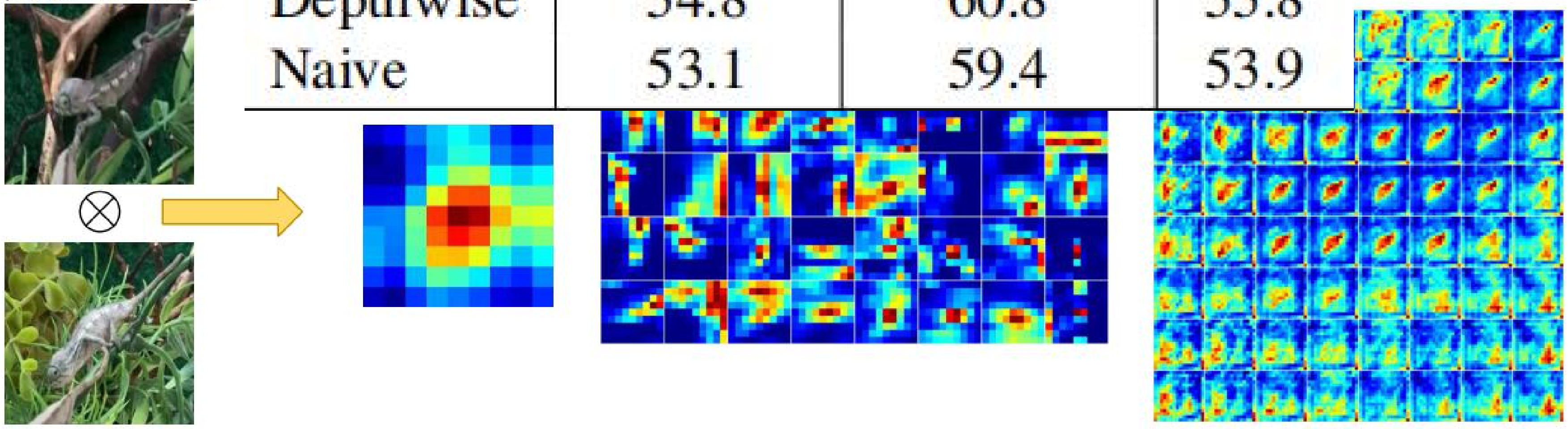
Alpha-Refine

Feature Fusion Options

- Naive Correlation or Depth-wise Correlation blur the response.
- Pixel-wise Correaltion maintain more spatial information, helpful to box estimation.



Method	AUC(%)	P_{Norm}(%)	P(%)
Pixelwise	**55.9**	**62.2**	**57.4**
Depthwise	54.8	60.8	55.8
Naive	53.1	59.4	53.9



(a) Reference image
(b) Test image
(c) Naive correlation responses
(d) Depth-wise correlation responses (the first 32 channels)
(e) Pixel-wise correlation responses (of all 64 pixel kernels)

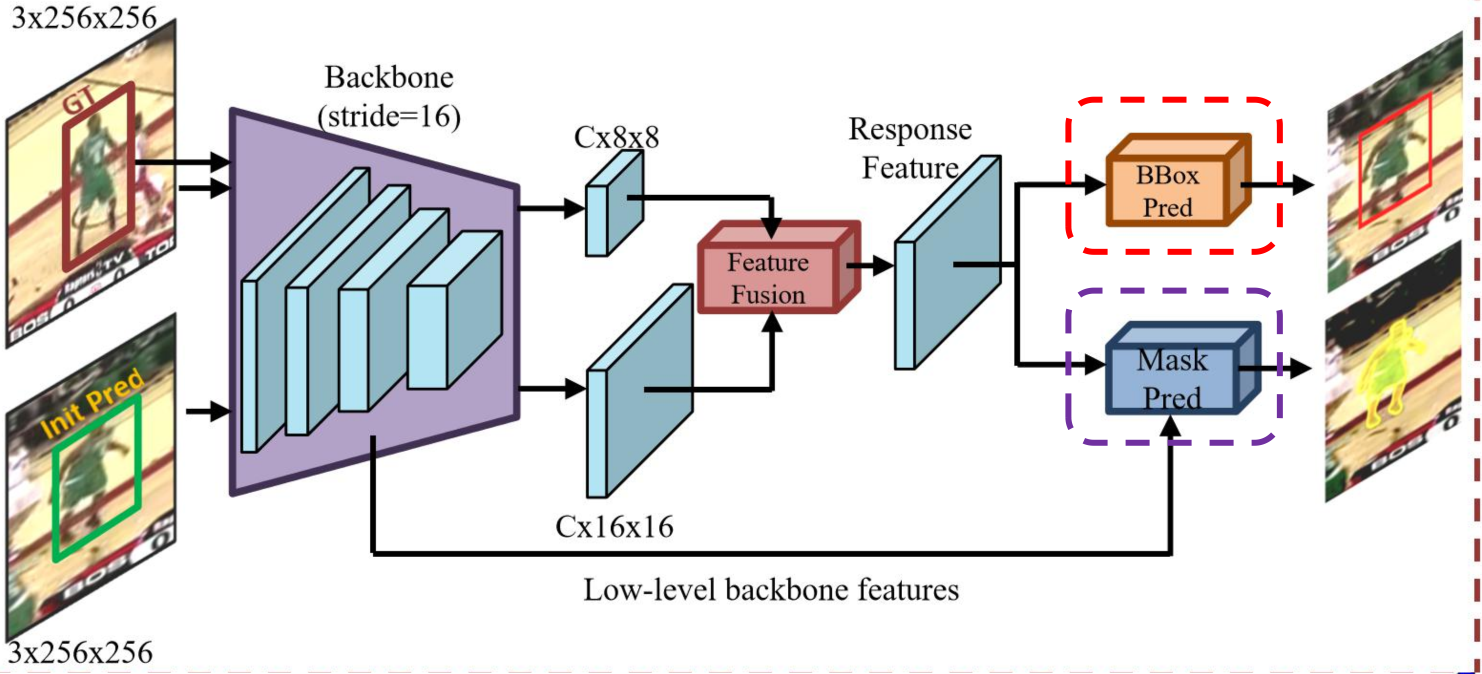
Alpha-Refine

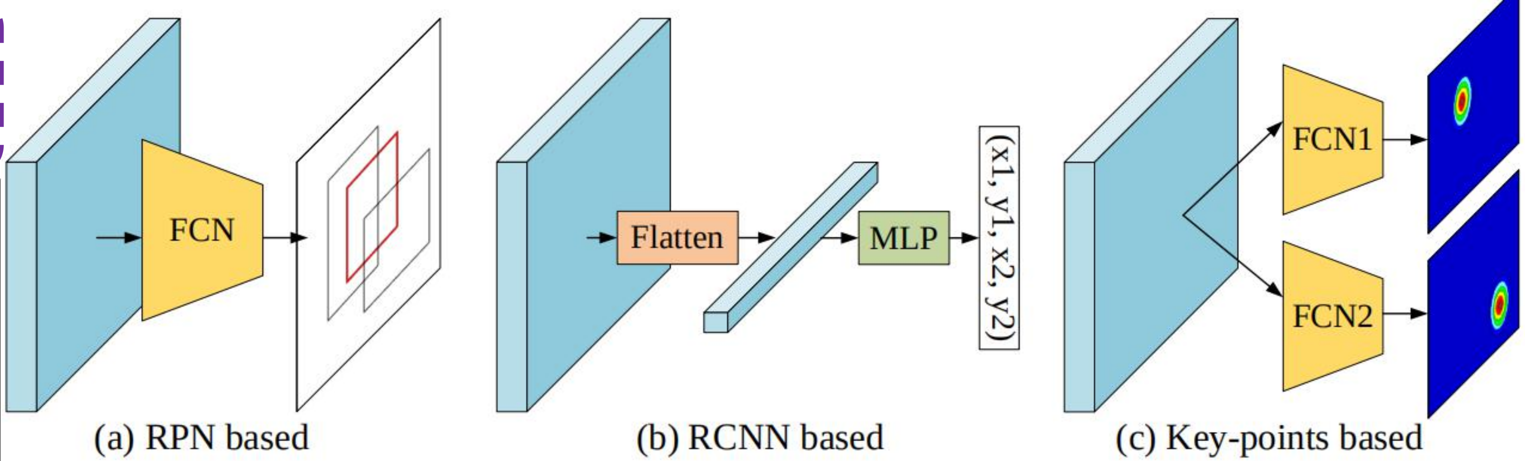
Head Options

- ➢ BBox Head
- ➢ Mask Head

Method	AUC(%)	P_{Norm}(%)	P(%)
SiamRPNpp	47.6	54.7	47.2
$+AR_{rpn}$	50.2	55.5	51.2
$+AR_{rcnn}$	48.9	54.2	46.9
$+AR_{c}$	54.6	60.3	55.3
$+AR_{rpn+m}$	53.7	60.3	54.7
$+AR_{rcnn+m}$	51.6	58.1	52.3
$+AR_{c+m}$	55.9	62.2	57.4

- ✓ Key-points-based Corner Head
- ✓ Auxiliary Mask Head





Alpha-Refine

Benchmark Result

Results on LaSOT

Method	Base			Base+AR		
	AUC	P_{Norm}	P	AUC	P_{Norm}	P
ECO	36.9	43.5	36.4	46.1	50.8	46.0
RT-MDNet	30.8	36.0	30.1	49.9	63.1	50.7
SiamRPNpp	47.6	54.7	47.2	55.9	62.2	57.4
ATOM	**49.5**	**56.0**	**49.1**	**57.0**	**63.0**	**58.1**
DiMP50	**55.9**	**63.3**	**55.3**	**60.2**	**66.8**	**61.7**
DiMPsuper	**63.7**	**72.5**	**65.6**	**65.3**	**73.2**	**68.0**

Results on GOT-10k

Method	Base			Base+AR		
	AO	$SR_{0.5}$	$SR_{0.75}$	AO	$SR_{0.5}$	$SR_{0.75}$
ECO	41.3	43.8	13.4	56.7	64.8	46.1
RT-MDNet	35.0	35.8	9.2	56.1	63.7	46.9
ATOM	**53.5**	**62.2**	**37.8**	**63.1**	**71.1**	**55.8**
SiamRPNpp	51.8	61.7	32.4	61.5	69.6	46.9
DiMP50	**60.3**	**71.8**	**46.0**	**65.4**	**74.3**	**58.5**
DiMPsuper	**67.2**	**78.8**	**59.3**	**70.1**	**80.0**	**64.2**

Results on TrackingNet

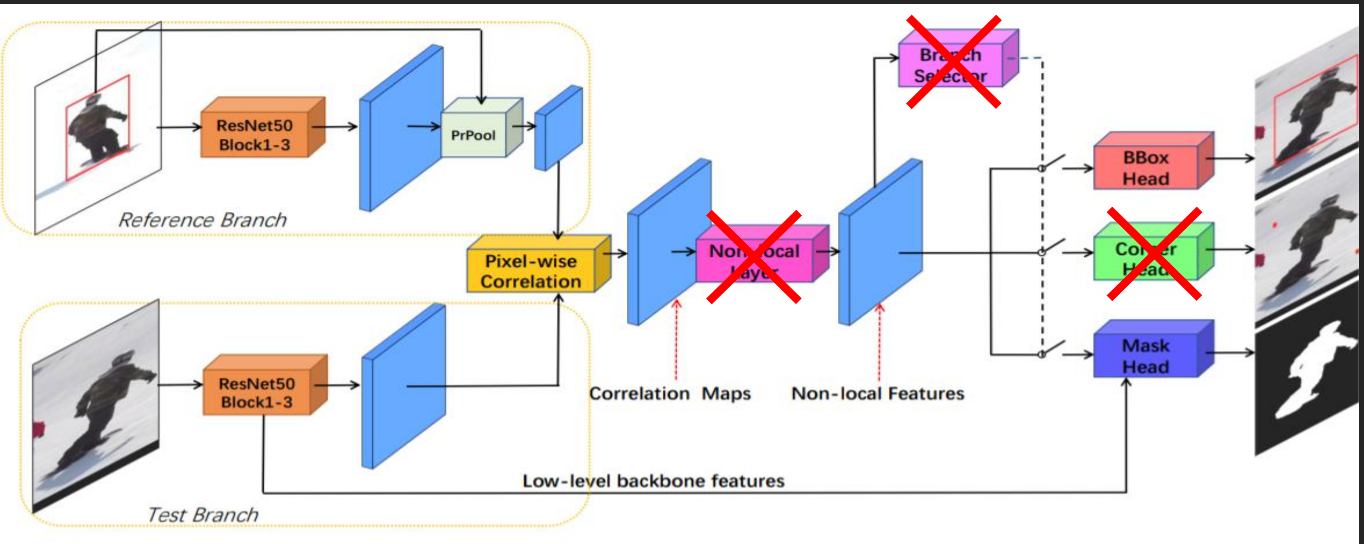
Method	Base			Base+AR		
	AUC	P_{Norm}	P	AUC	P_{Norm}	P
ECO	61.2	71.0	55.9	75.1	80.0	71.4
RT-MDNet	58.4	69.4	53.3	76.0	81.0	72.3
ATOM	70.3	77.1	64.8	77.7	82.5	74.5
SiamRPNpp	**73.3**	**80.0**	**69.4**	**78.8**	**83.7**	**76.4**
DiMP50	**74.0**	**80.1**	**68.7**	**79.5**	**84.1**	**76.5**
DiMPsuper	**77.6**	**82.5**	**72.6**	**80.5**	**85.6**	**78.3**

Results on VOT2020

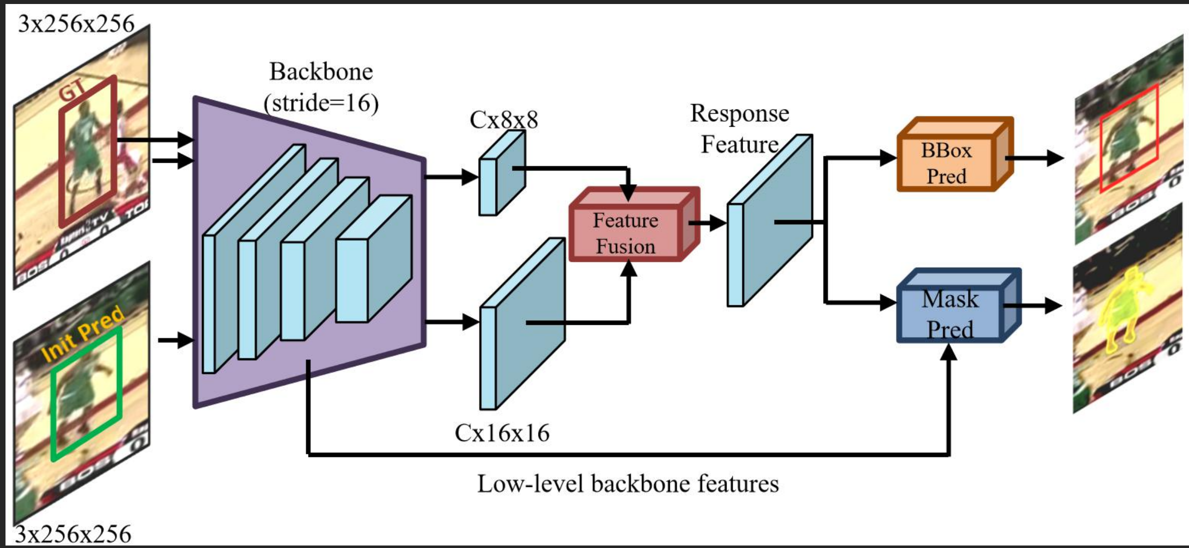
Method	Base		Base+AR	
	Baseline	Real Time	Baseline	Real Time
RT-MDNet	0.248	0.247	0.371	0.356
SiamRPNpp	0.254	0.254	0.395	0.395
ECO	**0.280**	**0.276**	**0.426**	**0.426**
ATOM	0.275	0.279	0.416	0.414
DiMP50	**0.286**	**0.278**	**0.444**	**0.438**
DiMPsuper	**0.314**	**0.311**	**0.471**	**0.478**

- Alpha-Refine significantly improve the base trackers under different benchmarks

Alpha-Refine



ECCV 2020 Reject



CVPR 2021 Accept

Bin Yan, **Xinyu Zhang**, Dong Wang, Huchuan Lu, Xiaoyun Yang. Alpha-Refine: Boosting Tracking Performance by Precise Bounding Box Estimation. CVPR, 2021.

- Code: https://github.com/MasterBin-IIAU/AlphaRefine

Alpha-Refine

■ **Reviewer #1：Weak accept**

…, **the proposed technique is indeed useful practically regardless of its weakness about novelty and heuristic algorithm design.** … **the practical benefit outweighs the limitations**, …

■ Reviewer #2：Weak accept

…**the paper has slightly limited novelty** (W1), **I believe that the contributions are significant enough for the tracking community to warrant acceptance.** **As also agreed by R3, the paper doesn't have a significantly novel idea and instead smartly combines existing ideas.** **However, the proposed approach is well motivated and thoroughly analysed**…**significant performance improvements** …

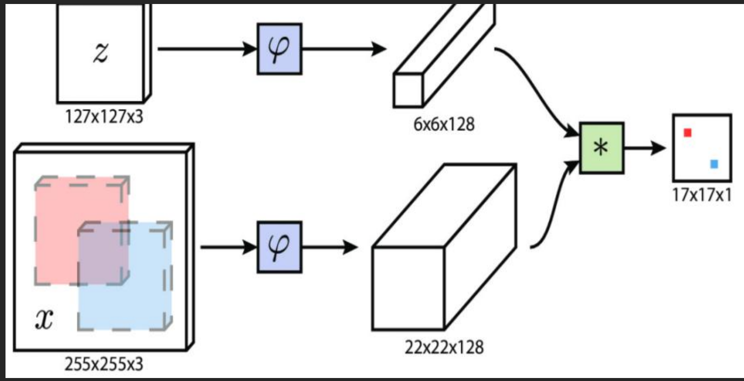
■ Reviewer #3：Strong accept

the strength of the proposal is more in a well-thought-out combination of existing ideas than in any especially new concept, **the performance of the proposed method is clearly and consistently superior to competing ones, and the added latency cost is modest.**

◆ **Meta-review**

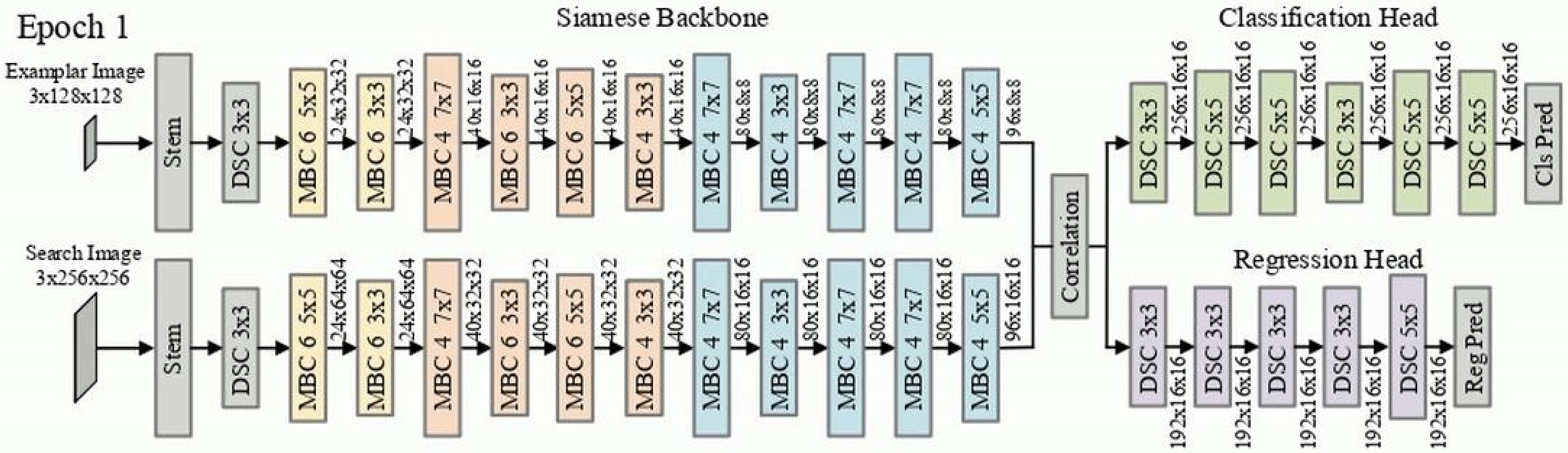
The consensus is clear - there is no major technical novelty, but the method is applicable to a broad range of trackers, **which show, sometimes significant, improvements if the alpha-refine module is attached. This is true even for strong baselines.**

SiamFC (ECCVW16)



LightTrack

LightTrack

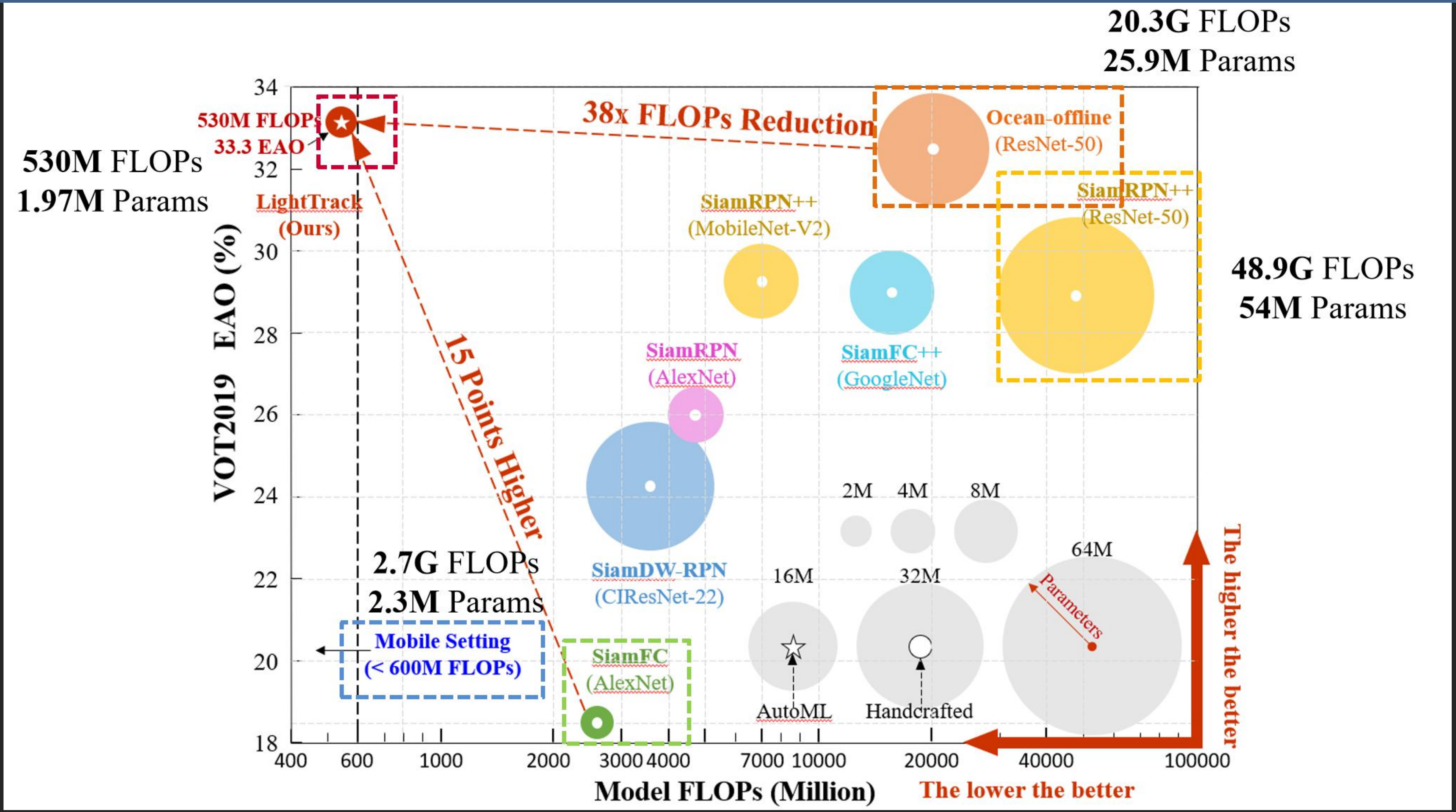


Bin Yan, Houwen Peng, Kan Wu, Dong Wang, Jianlong Fu, Huchuan Lu. LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search. CVPR, 2021.

- Code: https://github.com/researchmm/LightTrack

LightTrack

➢ Motivation

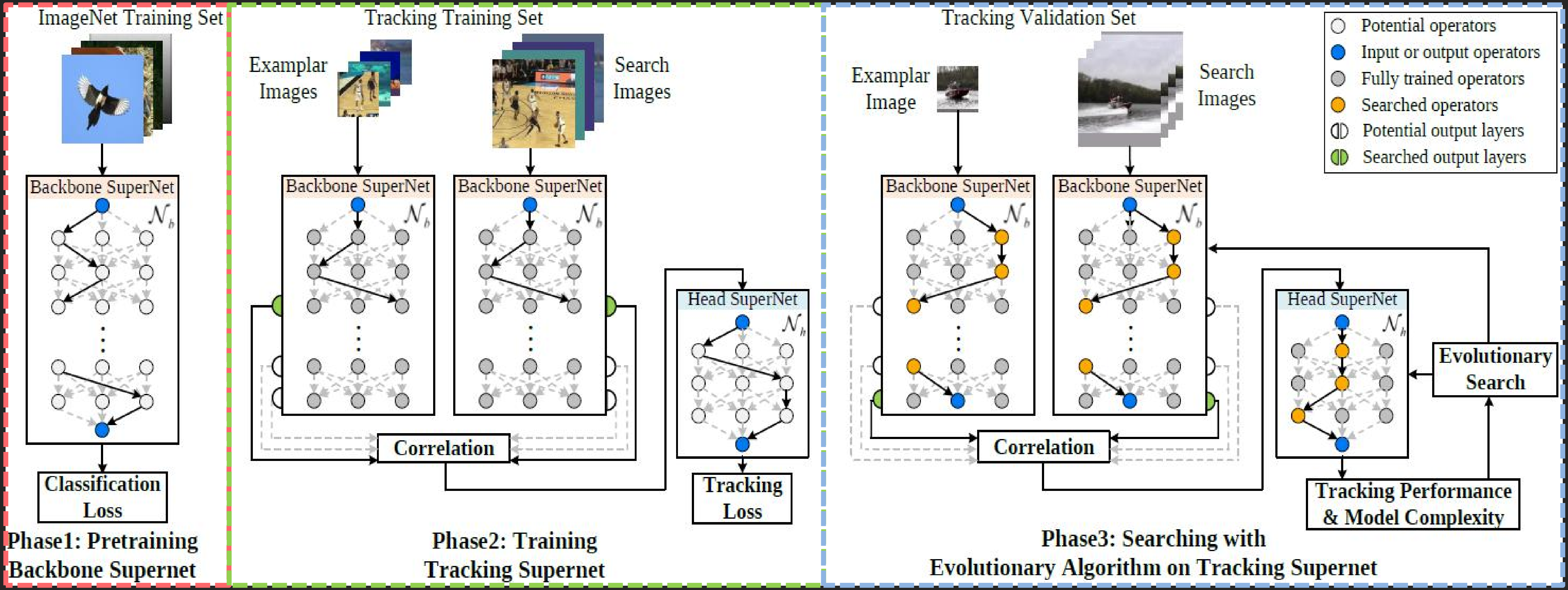


✓ SOTA object trackers are becoming increasingly heavy and expensive
✓ Tracking models are difficult to deploy in real-world applications

LightTrack

➢ Method

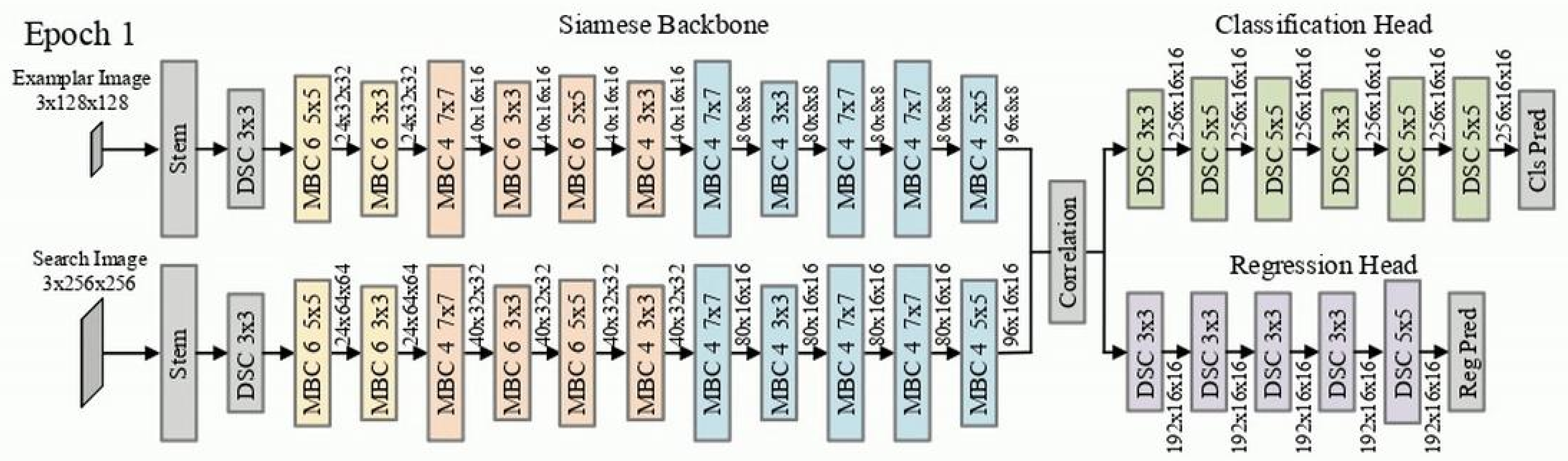
Framework (Pipeline)



LightTrack

- **Method**

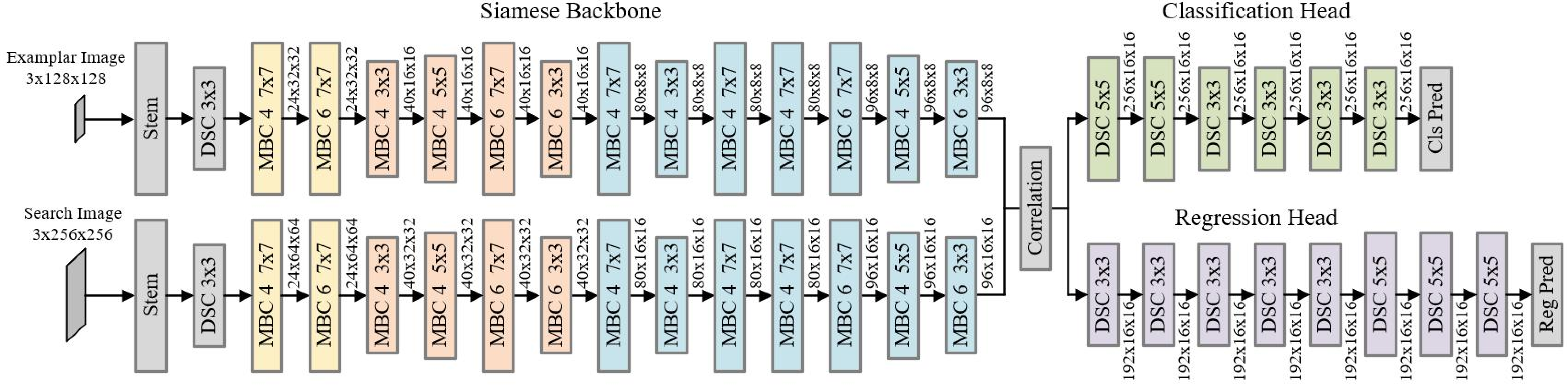
Search Process



LightTrack

➢ Method

Searched Architecture



- ✓ More than 50% layers in the backbone adopt kernel size 7x7 (large receptive fields can improve the localization precision)
- ✓ The searched architecture chooses the second-last block as the feature output layer. (higher-level feature might not be better)
- ✓ The classification branch contains fewer layers than the regression branch (coarse object localization is relatively easier than precise bounding box regression.

LightTrack

Comparison with SOTA Trackers

➢VOT2019

	SiamMask [50]	SiamFC++(G) [52]	SiamRPN++(M) [30]	ATOM [14]	TKU [48]	DiMP^{r} [6]	Ocean(off) [56]	**Ours**
EAO(↑)	0.287	0.288	0.292	0.301	0.314	**0.321**	**0.327**	**0.333**
Accuracy(↑)	**0.594**	0.583	0.580	**0.603**	0.589	0.582	**0.590**	0.536
Robustness(↓)	0.461	0.406	0.446	0.411	**0.349**	**0.371**	0.376	**0.321**
FLOPs(G)(↓)	**15.5**	17.5	**7.0**	-	-	-	20.3	**0.53**
Parameters(M)(↓)	16.6	13.9	**11.2**	**8.4**	-	26.1	25.9	**1.97**

➢GOT-10K

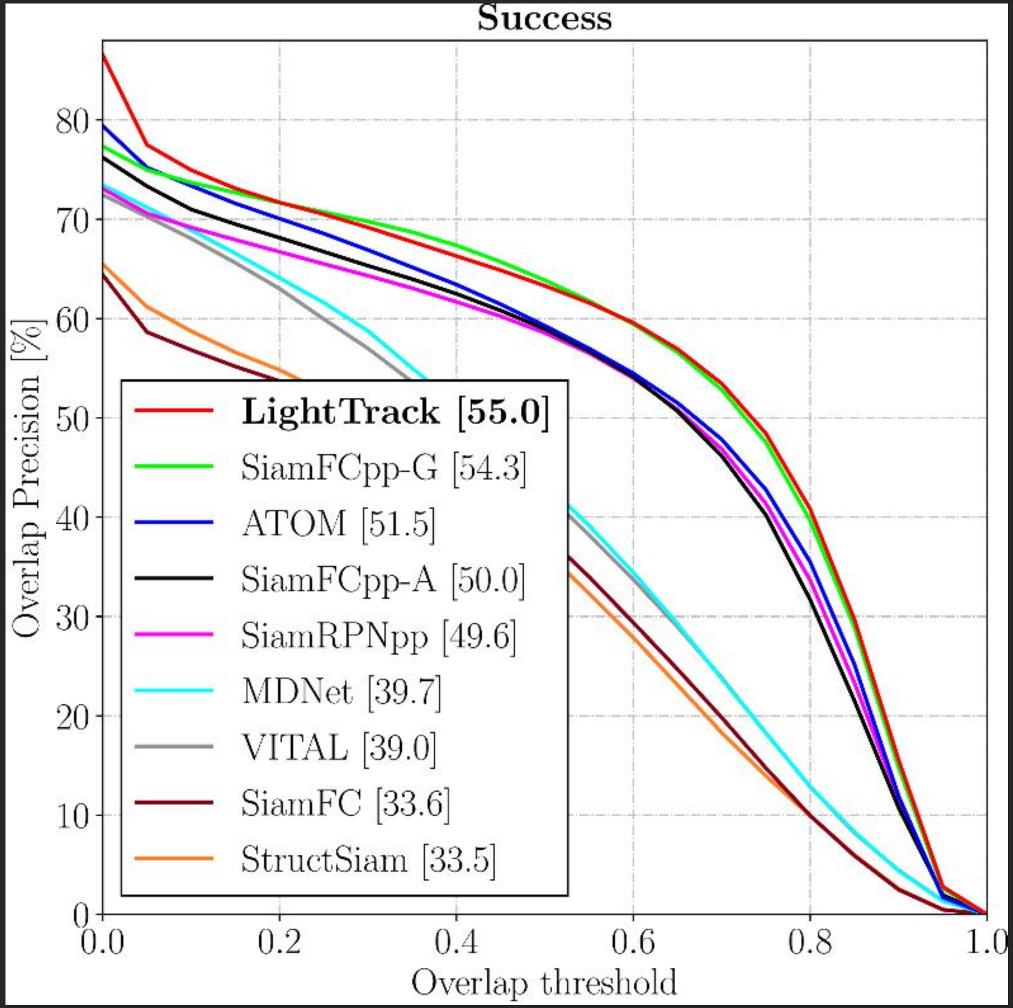
	DaSiam [57]	SiamRPN++(R) [30]	ATOM [14]	Ocean-offline [56]	SiamFC++(G) [52]	Ocean-online [56]	DiMP-50 [6]	**Ours**
AO(↑)	0.417	0.518	0.556	**0.592**	**0.595**	**0.611**	**0.611**	**0.611**
SR0.5(↑)	0.461	0.618	0.634	0.695	0.695	**0.721**	**0.712**	**0.710**
FLOPs(G)(↓)	**21.0**	48.9	-	20.3	**17.5**	-	-	**0.53**
Parameters(M)(↓)	19.6	54.0	**8.4**	25.9	**13.9**	44.3	26.1	**1.97**

➢TrackingNet

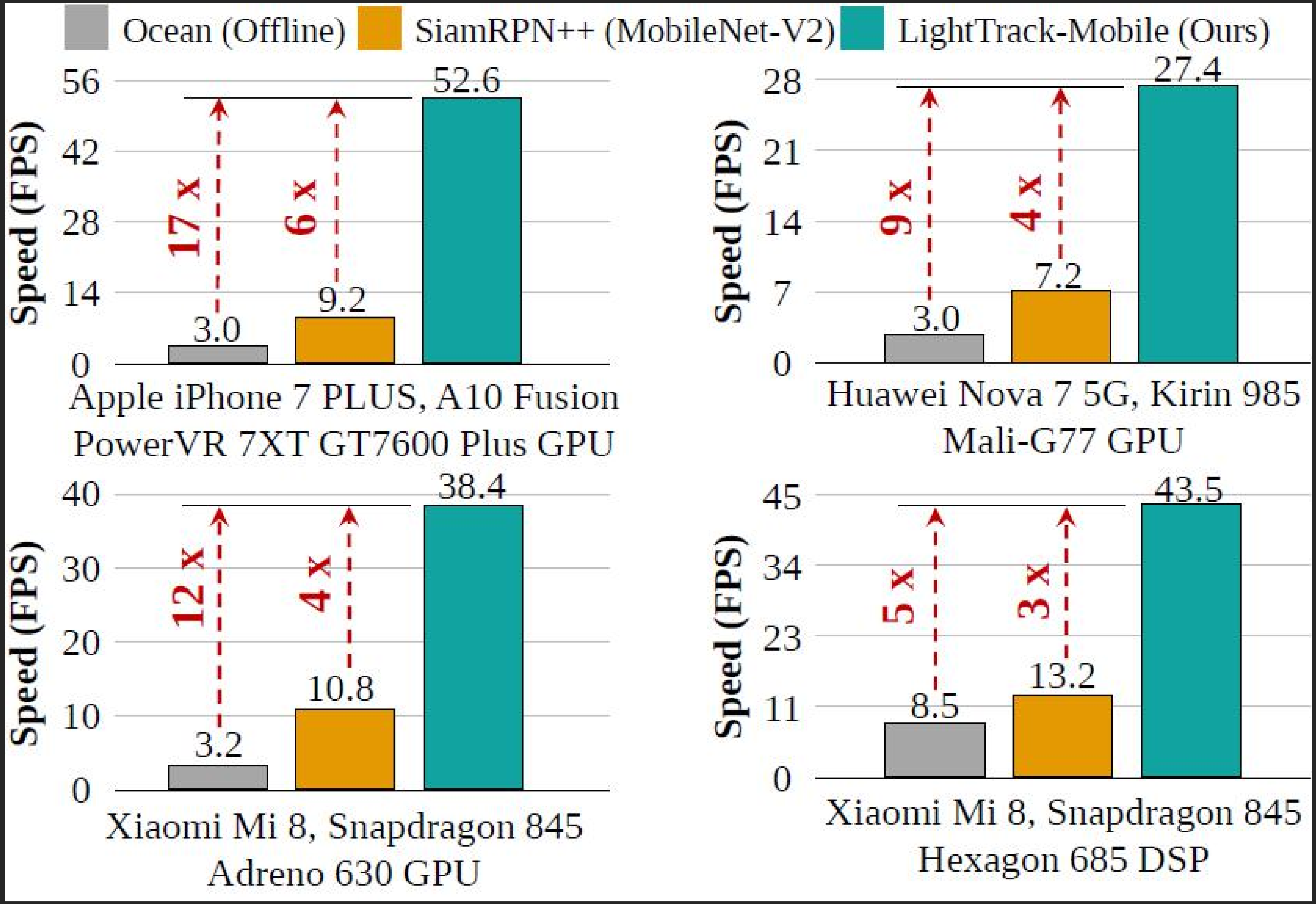
	RTMDNet [27]	ECO [13]	DaSiam [57]	C-RPN [17]	ATOM [14]	SiamFC++(A) [52]	SiamRPN++(R) [30]	DiMP-50 [6]	**Ours**
P(%)	53.3	55.9	59.1	61.9	64.8	64.6	**69.4**	**68.7**	**69.5**
P_{norm}(%)	69.4	71.0	73.3	74.6	77.1	75.8	**80.0**	**80.1**	**77.9**
AUC(%)	58.4	61.2	63.8	66.9	70.3	71.2	**73.3**	**74.0**	**72.5**

LightTrack

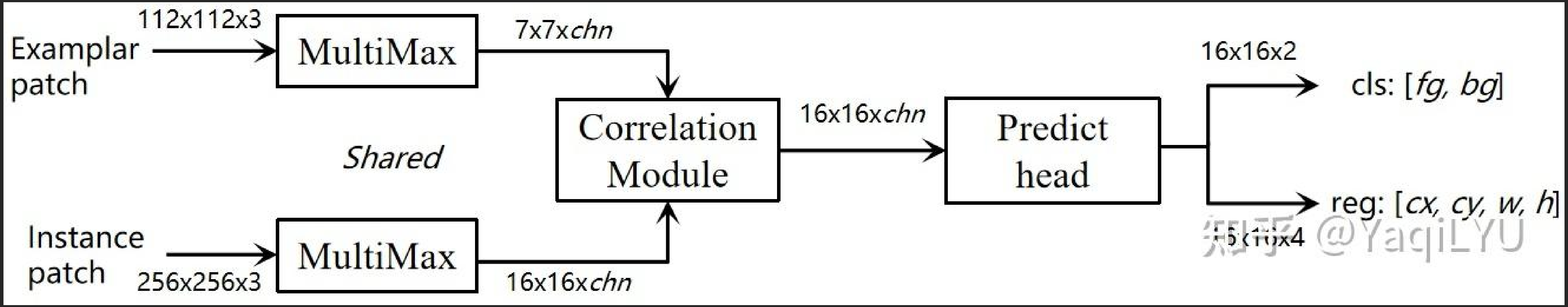
LaSOT



Speed on Resource-Limited Platforms



LightTrack



Models	Source	Size(M)	FLOPs(G)	AUC	P_{norm}	P
SiamRPN++(MobileNetv2)	CVPR2019	-	-	45.0	52.5	45.1
Ocean-offline	ECCV2020	25.9	-	52.6	-	52.6
LightTrack-LargeB	CVPR2021	3.13	0.79	55.5	-	56.1
LightTrack-LargeA	CVPR2021	2.62	0.78	55.0	-	55.2
LightTrack-Mobile	CVPR2021	1.97	0.53	53.8	-	53.7
Our-nomal	-	**1.11**	**0.44**	**57.2**	**65.6**	**57.7**
Our-light	-	**0.29**	**0.12**	**52.4**	**62.2**	**52.3**

Model	iPhone11 (A13) CoreML-ANE	Mi11(Snapdragon 888) MNN-CPU
Our-nomal	72.5 FPS	47.2 FPS
Our-light	175.4 FPS	140.8 FPS

转自：

YaqiLYU

https://zhuanlan.zhihu.com/p/419900331
让视觉目标跟踪在移动端起飞(三)

High-Performance Visual Tracking Algorithms

■ Conclusion

All codes can be found here!

https://github.com/wangdongdut/Online-Visual-Tracking-SOTA

- ➢ A General and Robust Meta-Update Scheme
- ➢ A New Transformer-based Tracking Framework
- ➢ A General and Accurate Scale Estimation Module
- ➢ A Flexible Light-Weight Tracking Method

Thanks!