Towards Sequence-Level Training for Visual Tracking

Minji Kim[1]*, Seungkwan Lee[2,3]*, Jungseul Ok[2], Bohyung Han[1], Minsu Cho[2]

* denotes equal contribution

Code

ECCV TEL AVIV 2022

Background

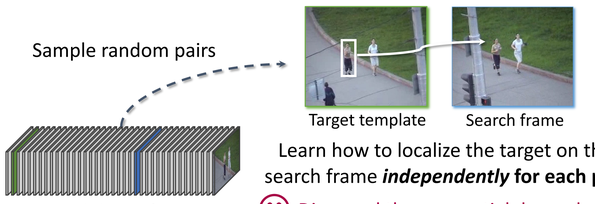
Visual Object Tracking (VOT)

- Given a video $\mathbf{v} = (v_0, \dots, v_T)$ of $T+1$ frames and the target state (e.g., bbox) l_0 in the first frame v_0,

- VOT aims to ***sequentially*** predict the target states $l_1 \sim l_T$ in the subsequent frames $v_1 \sim v_T$.

- Most trackers predict the current state l_t based on the previous prediction l_{t-1} and the current observation v_t.
- The objective of a tracking algorithm is to maximize *the **sequence-level*** performance $r(\mathbf{l})$, where $\mathbf{l} = (l_1, \dots, l_T)$ and r is a performance metric.

Typical approach: Frame-Level Training (FLT)

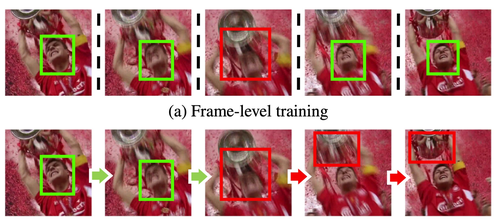


Learn how to localize the target on the search frame ***independently*** **for each pair**

☹ Disregard the sequential dependency

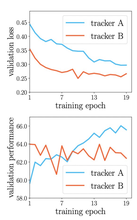
Motivation

Problem: training-testing inconsistency in FLT



(a) Frame-level training

(b) Sequence-level self-testing



(c) Loss *vs.* Perf.

- FLT (a) does not necessarily improve actual tracking (b).
- Inconsistency between the loss and the perf. is often observed as in (c).

What causes such inconsistency?

	Testing	Frame-Level Training
1) Data Distributions	Search window is determined by ***previous*** **estimation**	Search window is determined by **GT + random perturbation**
2) Task Objectives	Retaining successful localization **over a sequence**	**Immediate** localization quality in each frame

Our Approach

Main idea: Sequence-Level Training (SLT) based on reinforcement learning

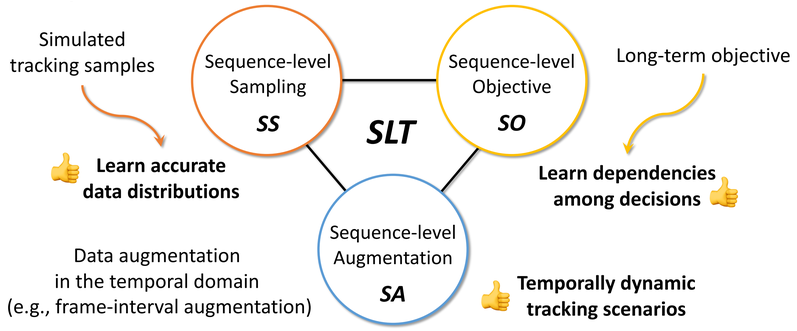
- Training a model by ***actually tracking*** a target on a video and ***directly optimizing*** the sequence-level performance $r(\mathbf{l})$.
- Training objective: $L(\theta) := -\mathbb{E}_{\mathbf{l}\sim\pi_\theta}[r(\mathbf{l})]$
- How to minimize $L(\theta)$?
  - Compute the gradient using the REINFORCE algorithm:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{\mathbf{l}\sim\pi_\theta}[r(\mathbf{l})\nabla_\theta \log p_\theta(\mathbf{l})]$$

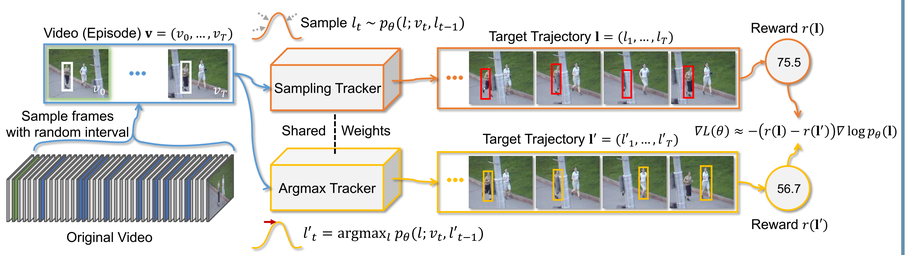
  - For each training sequence, the gradient is approximated as follows:

$$\nabla_\theta L(\theta) \approx -r(\mathbf{l})\nabla_\theta \log p_\theta(\mathbf{l})$$

Three components of SLT



Self-critical SLT



- To reduce the variance of gradient estimation, we adopt the self-critical sequence training (SCST).
- In training time, there are two trackers sharing network parameters: a sampling tracker and an argmax tracker.
- For each training step, a video is played twice independently by both trackers.
- A reward from the argmax tracker is used as a baseline reward to train the sampling tracker.

Integration into recent trackers: SiamRPN++, SiamAttn, TrDiMP, TransT

- Our training method assumes the target localization is a ***stochastic*** action.
- Recent trackers typically include a greedy box selection procedure, where the most confident box among candidates is selected.
- We convert the greedy box selection to become stochastic.
  - Confidence scores of N candidates → a categorical distribution of N categories.
- Then, we reinforce the anchor selection procedure using the proposed SLT.
- Before SLT, we pre-train the trackers using their original frame-level training methods.

Evaluation

Effect of SLT with four baseline trackers on three benchmarks

Method		LaSOT AUC (Δ)	LaSOT P_{Norm}	TrackingNet AUC (Δ)	TrackingNet P_{Norm}	TrackingNet P	GOT-10k AO (Δ)	GOT-10k $SR_{0.5}$	GOT-10k $SR_{0.75}$
SiamRPN++	Base	51.0	60.3	68.2	78.3	68.9	49.5	58.0	30.5
	+SLT	58.4 (+7.4)	66.6	75.8 (+7.6)	81.0	71.3	62.1 (+12.6)	74.9	49.0
SiamAttn	Base	54.8	63.5	74.3	80.9	70.6	53.4	61.8	36.4
	+SLT	57.4 (+2.6)	66.2	76.9 (+2.6)	82.3	72.6	62.5 (+9.1)	75.4	50.2
TrDiMP	Base	63.3	72.3	78.1	83.3	73.1	67.1	77.4	58.5
	+SLT	64.4 (+1.1)	73.5	78.1 (+0.0)	83.1	73.1	67.5 (+0.4)	78.8	58.7
TransT	Base	64.2	73.7	81.1	86.8	80.1	66.2	75.5	58.7
	+SLT	66.8 (+2.6)	75.5	82.8 (+1.7)	87.5	81.4	67.5 (+1.3)	76.5	60.3

Comparison with SOTA trackers on LaSOT

	PACNet [46]	Ocean [48]	DiMP50 [2]	PrDiMP50 [8]	TransT [4]	STARK-ST50 [42]	STARK-ST101 [42]	**SLT-SiamRPN++**	**SLT-SiamAttn**	**SLT-TrDiMP**	**SLT-TransT**
AUC (%)	55.3	56.0	56.9	59.8	64.2	66.4	**67.1**	58.4	57.4	64.4	66.8
P_{Norm} (%)	62.8	65.1	64.3	68.0	73.7	76.3	**77.0**	66.6	66.2	73.5	75.5

Comparison with SOTA trackers on TrackingNet

	DiMP50 [2]	SiamFC++ [41]	MAML [35]	PrDiMP50 [8]	TransT [4]	STARK-ST50 [42]	STARK-ST101 [42]	**SLT-SiamRPN++**	**SLT-SiamAttn**	**SLT-TrDiMP**	**SLT-TransT**
AUC (%)	74.0	75.4	75.7	75.8	81.1	81.3	82.0	75.8	76.9	78.1	**82.8**
P_{Norm} (%)	80.1	80.0	82.2	81.6	86.8	86.1	86.9	81.0	82.3	83.1	**87.5**

Comparison with SOTA trackers on GOT-10k

	Add. data	SiamFC++ [41]	DiMP50 [2]	Ocean [48]	PrDiMP50 [8]	TransT [4]	TrDiMP [36]	STARK-ST50 [42]	**SLT-SiamRPN++**	**SLT-SiamAttn**	**SLT-TrDiMP**	**SLT-TransT**
AO (%)		59.5	61.1	61.1	63.4	66.2	67.1	**68.0**	62.1	62.5	67.5	67.5
$SR_{0.5}$ (%)	-	69.5	71.7	72.1	73.8	75.5	77.4	77.7	74.9	75.4	**78.8**	76.5
$SR_{0.75}$ (%)		47.9	49.2	47.3	54.3	58.7	58.5	**62.3**	49.0	50.2	58.7	60.3
AO (%)	✓	-	60.4	-	65.2	71.9	68.6	71.5	56.9	62.8	69.0	**72.5**

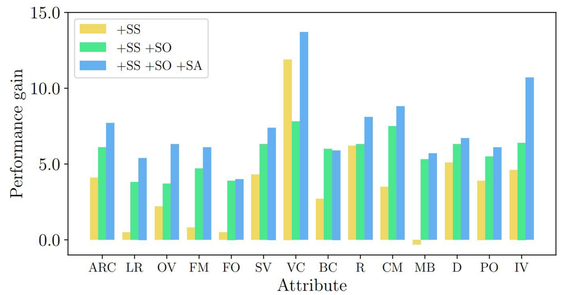
Analysis

Effect of SLT components

Baseline: SiamRPN++

Benchmark	SiamRPN++ Baseline	+SS (Δ)	+SS+SO (Δ)	+SS+SO+SA (Δ)
LaSOT (AUC)	51.0	55.1 (+4.1)	57.3 (+6.3)	58.4 (+7.4)
TrackingNet (AUC)	68.2	73.5 (+5.3)	75.0 (+6.8)	75.8 (+7.6)
GOT-10k (AO)	66.4	70.2 (+3.8)	73.8 (+7.4)	74.3 (+7.9)

Attribute analysis on LaSOT



ARC: Aspect Ratio Change
LR: Low Resolution
OV: Out-of-View
FM: Fast Motion
FO: Full Occlusion
SV: Scale Variation
VC: Viewpoint Change
BC: Background Clutter
R: Rotation
CM: Camera Motion
MB: Motion Blur
D: Deformation
PO: Partial Occlusion
IV: Illumination Variation

Effect of sequence-level data augmentation (SA)

Method	SA	GOT-10k (AO) $i=1$	$i=2$	$i=3$	LaSOT (AUC) $i=1$	$i=2$	$i=3$	$i=4$
SiamRPN++	-	66.4	63.1	60.8	51.0	50.0	50.2	48.8
SLT-SiamRPN++	-	73.8	67.9	65.5	57.3	55.1	54.1	52.6
SLT-SiamRPN++	✓	74.3	70.8	67.8	58.4	56.9	56.2	54.6

Summary

- We propose a novel sequence-level training strategy for visual tracking to resolve the training-testing inconsistency problem of recent trackers.
- Unlike existing methods, it trains a tracker by actually tracking on a video and directly optimizing a test-time performance metric.
- Experiments on four representative trackers demonstrate its effectiveness in learning visual tracking.