Towards Sequence-Level Training for Visual Tracking

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

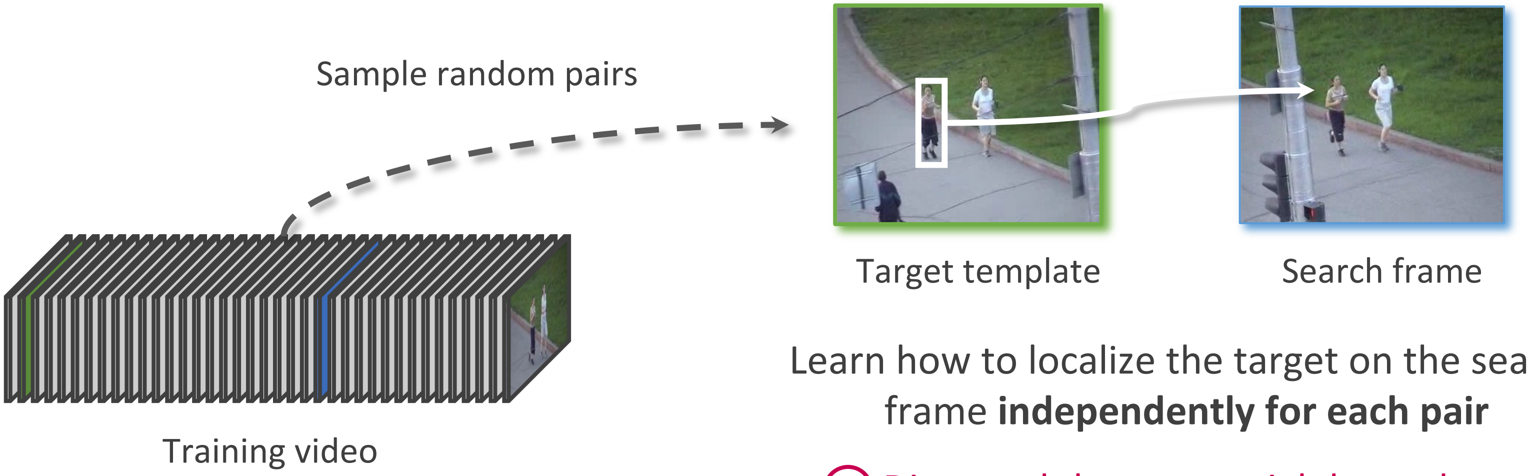
[1] Seoul National University
[2] POSTECH
[3] Deeping Source Inc.

* denotes equal contribution

ECCV TEL AVIV 2022

Visual Object Tracking (VOT)

- Given the target state (e.g., box) in the first frame, VOT aims to predict the target state in the subsequent frames
- Recent paradigm: **Frame-Level Training (FLT)**



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

☹ Disregard the sequential dependency

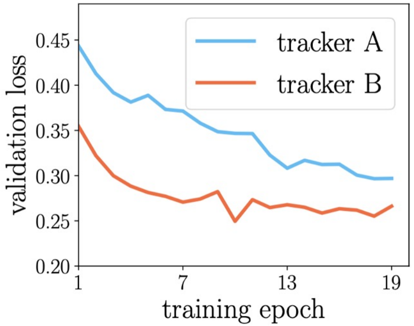
Pitfall of Frame-Level Training

- FLT does not necessarily improve the actual tracking performance

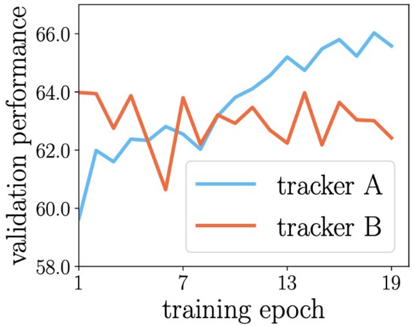
Training / Testing Inconsistency

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

Training / Testing Inconsistency



Tracker A gives a higher loss according to the frame-level objective

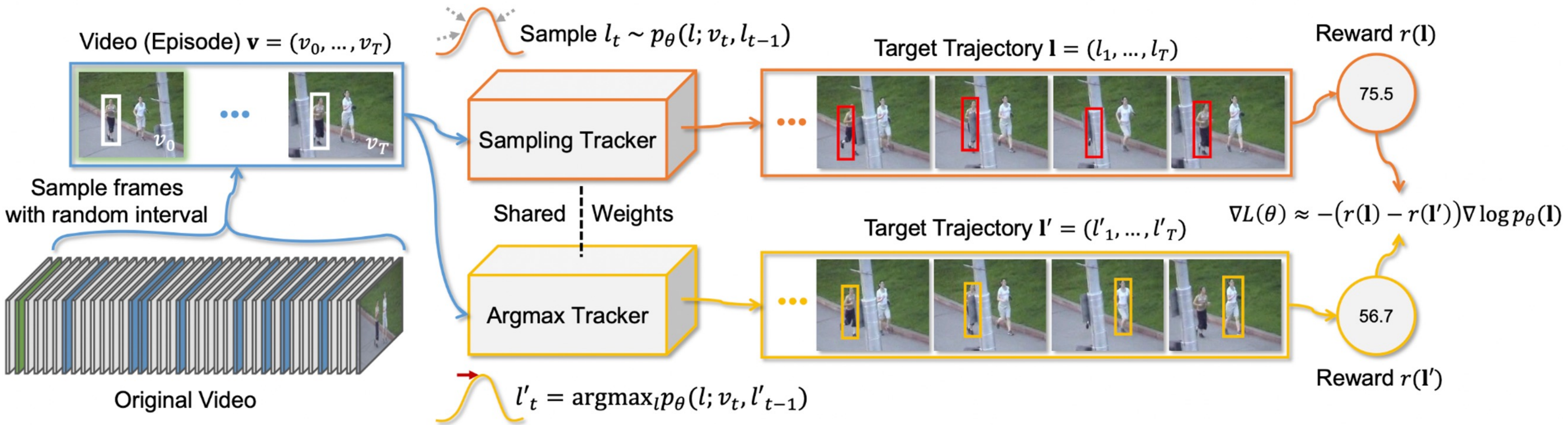


After 10 epochs, tracker A outperforms tracker B in terms of sequence-level performance

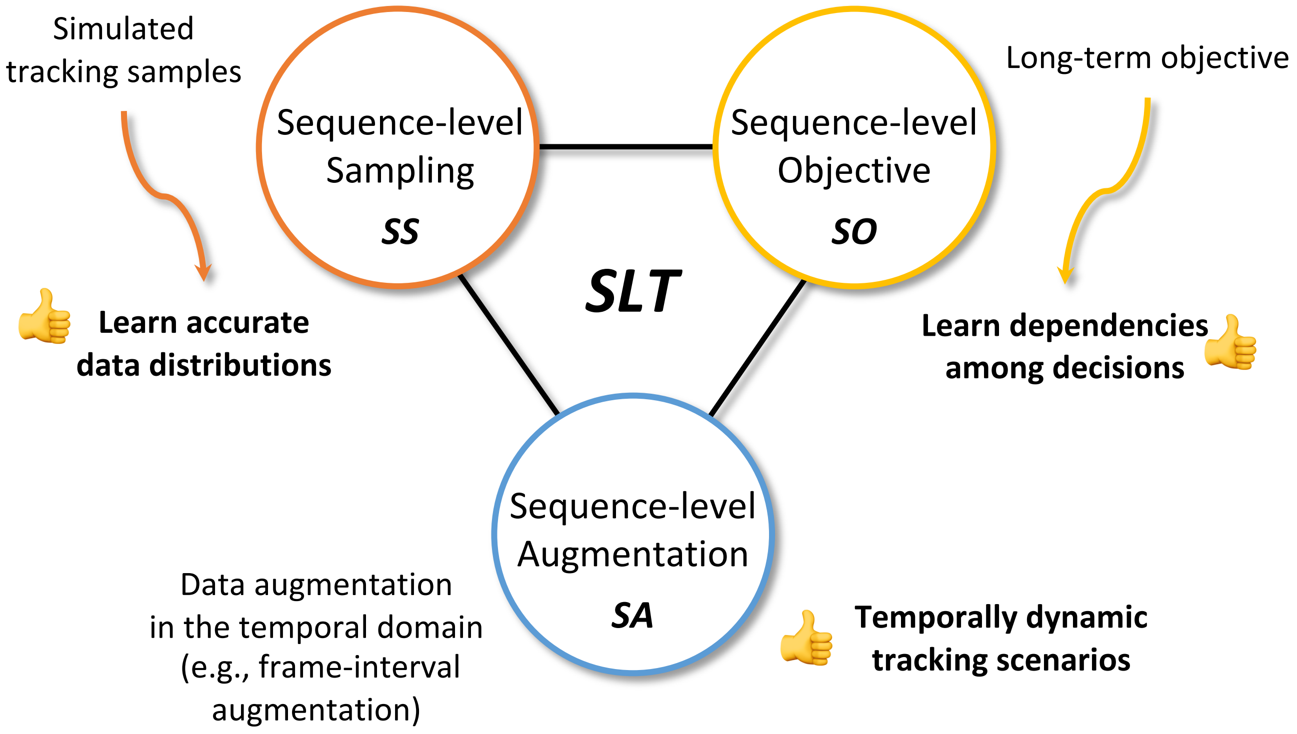
☹ Mismatch between validation loss/performance

Sequence-Level Training (SLT)

- Goal: to resolve the training / testing inconsistency in recent trackers
- Sequence-Level Training (SLT)
  - Based on reinforcement learning
  - Train a model by **actually tracking** on a video and **directly optimizing** a test-time metric



Sequence-Level Training (SLT)



Sequence-Level Training (SLT)

- Problem definition
  - Given a video $\mathbf{v} = (v_0, \ldots, v_T)$ and GT box g_0 of frame v_0,
    a tracker sequentially predicts a bounding box l_t of the target in each frame

prediction → $$l_t = \pi_\theta(o_t)$$ ← observation

tracker (agent)

→ $$\mathbf{l} = (l_1, \ldots, l_T)$$

Sequence of decisions

  - Objective of tracking : **maximize a *sequence-level* performance** $r(\mathbf{l})$

Sequence-Level Training (SLT)

- Idea: directly optimize the real objective of tracking
  - Minimize the negative expected reward:

sequence-level performance

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

data samples (= tracking trajectories)
from a tracker

  - Approximate the gradient with REINFORCE algorithm:

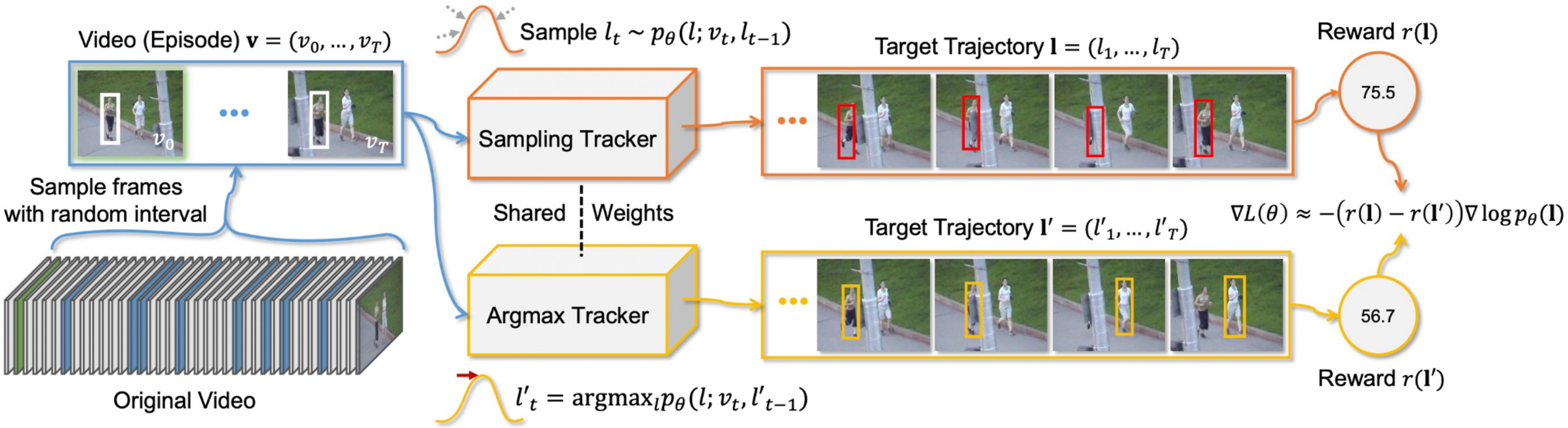
$$\nabla_\theta L(\theta) = -\mathbb{E}_{\mathbf{l}\sim\pi_\theta}[r(\mathbf{l})\nabla_\theta \log p_\theta(\mathbf{l})] \longrightarrow \nabla_\theta L(\theta) \approx -r(\mathbf{l})\nabla_\theta \log p_\theta(\mathbf{l})$$

Sequence-Level Training (SLT)

- Self-critical sequence training
  - To reduce a variance of gradient estimation
  - Exploit the test-mode performance of the current model as a baseline for the reward

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

Sequence-Level Training (SLT)



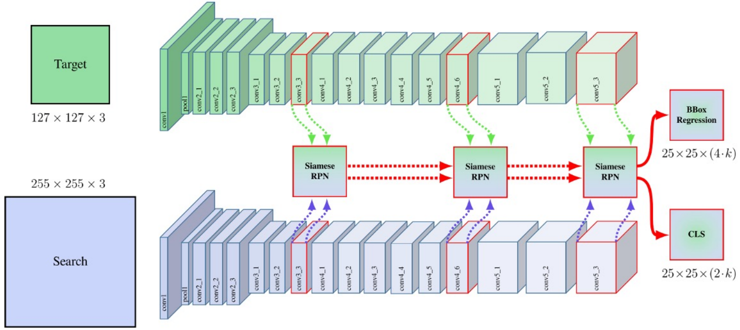
Reward from sampling mode

Reward from argmax mode

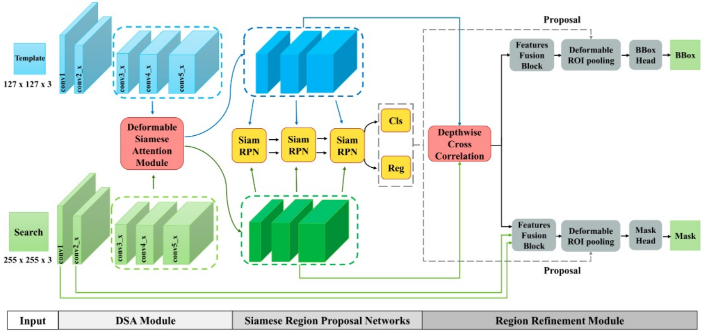
$$\nabla_\theta L(\theta) \approx -(r(\mathbf{l}) - r(\mathbf{l}'))\nabla_\theta \log p_\theta(\mathbf{l})$$

Self-critical reward

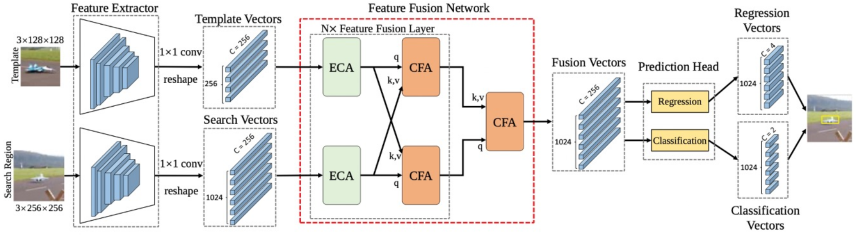
Integration into Tracking Algorithms



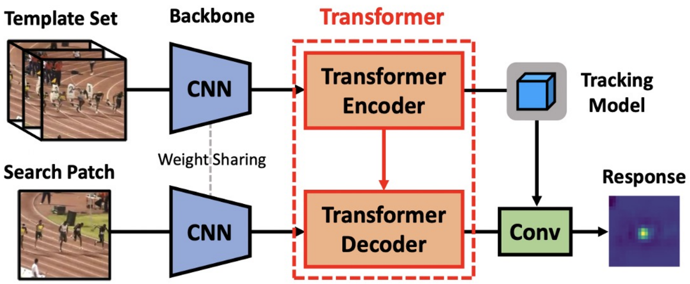
SiamRPN++ [CVPR19]



SiamAttn [CVPR20]



TransT [CVPR21]



TrDiMP [CVPR21]

Integration into Tracking Algorithms

Our training method assumes the target localization is a ***stochastic*** action

$$p(n) = \frac{\exp(\sigma^{-1}(x_n))}{\sum_{m=1}^{N} \exp(\sigma^{-1}(x_m))} \longrightarrow L = -(r(\mathbf{l}) - r(\mathbf{l}')) \sum_{t=1}^{T} \log p(n_t)$$

Convert *greedy* target selection to become *stochastic*

Plug the self-critical loss into the classification branch

$$L_{\text{siamrpn++}} = L + L_{\text{bbox}}$$

$$L_{\text{siamattn}} = L + \lambda_1 L_{\text{bbox}} + \lambda_2 L_{\text{refine-bbox}} + \lambda_3 L_{\text{mask}}$$

$$L_{\text{transt}} = L + \lambda_4 L_{\text{bbox-L1}} + \lambda_5 L_{\text{bbox-GIoU}}$$

$$L_{\text{trdimp}} = L + \lambda_6 L_{\text{iou-net}}$$

Experiments

Table 1: Performance of sequence-level training on LaSOT, TrackingNet, and GOT-10k.

Method		LaSOT		TrackingNet			GOT-10k		
		AUC (Δ)	P_{Norm}	AUC (Δ)	P_{Norm}	P	AO (Δ)	$SR_{0.5}$	$SR_{0.75}$
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

Experiments

Table 2: Comparison with the state-of-the-art 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

Table 3: Comparison with the state-of-the-art 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**

Table 4: Comparison with the state-of-the-art trackers on GOT-10k. 'Add. data' denotes that trackers are trained using additional training datasets other than 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**

Ablation

- Sequence-level Sampling (SS)
  - Robust to variations of aspect ratio, scale, rotation, illumination
- Sequence-level Objective (SO)
  - Prevent the tracker to lose the target in challenging situations such as full occlusion, background clutters, motion blur
- Sequence-level Augmentation (SA)
  - Boosts the overall performance

Table 5: Effect of sequence-level training components.

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)

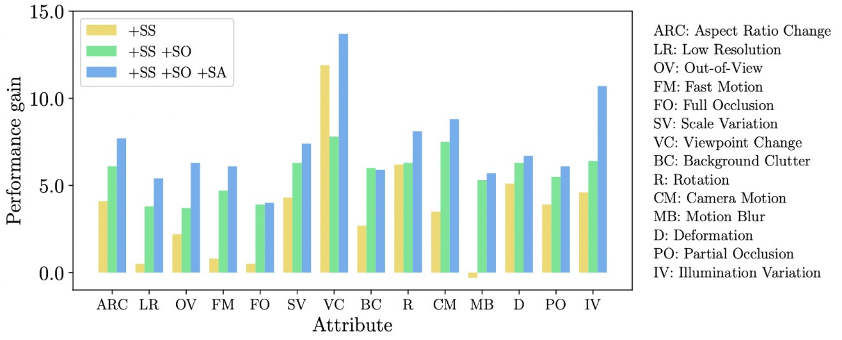


Fig. 3: Benefits of sequence-level training components to individual attributes on the LaSOT dataset. The baseline tracker is SiamRPN++, and the y-axis is performance (AUC) gain compared with the baseline tracker.

Thank you

https://github.com/byminji/SLTtrack