CHASE: Robust Visual Tracking via Cell-Level Differentiable Neural Architecture Search

Seyed Mojtaba Marvasti-Zadeh*†[1,2]
mojtaba.marvasti@ualberta.ca
Javad Khaghani*[1]
khaghani@ualberta.ca
Li Cheng[1]
lcheng5@ualberta.ca
Hossein Ghanei-Yakhdan[2]
hghaneiy@yazd.ac.ir
Shohreh Kasaei[3]
kasaei@sharif.edu

[1] Vision and Learning Lab,
University of Alberta,
Edmonton, Canada
[2] Digital Image & Video Processing Lab,
Yazd University,
Yazd, Iran
[3] Image Processing Lab,
Sharif University of Technology,
Tehran, Iran

Abstract

A strong visual object tracker nowadays relies on its well-crafted modules, which typically consist of manually-designed network architectures to deliver high-quality tracking results. Not surprisingly, the manual design process becomes a particularly challenging barrier, as it demands sufficient prior experience, enormous effort, intuition, and perhaps some good luck. Meanwhile, neural architecture search has gaining grounds in practical applications as a promising method in tackling the issue of automated search of feasible network structures. In this work, we propose a novel cell-level differentiable architecture search mechanism with early stopping to automate the network design of the tracking module, aiming to adapt backbone features to the objective of Siamese tracking networks during offline training. Besides, the proposed early stopping strategy avoids over-fitting and performance collapse problems leading to generalization improvement. The proposed approach is simple, efficient, and with no need to stack a series of modules to construct a network. Our approach is easy to be incorporated into existing trackers, which is empirically validated using different differentiable architecture search-based methods and tracking objectives. Extensive experimental evaluations demonstrate the superior performance of our approach over five commonly-used benchmarks.

1 Introduction

Visual object tracking (VOT) aims to localize an unknown object in sequential video frames, just given its initial state. Visual trackers constantly seek to find more robust and accurate approaches considering various applications and challenges in real-world scenarios. In the spirit of *deep learning* (DL), an important objective is to design reliable network architectures for visual tracking purposes [28], usually requiring adequate experience, insightful knowledge, learning heuristics, and extensive manual trial & error.

*These authors contributed equally to this work
† Corresponding author

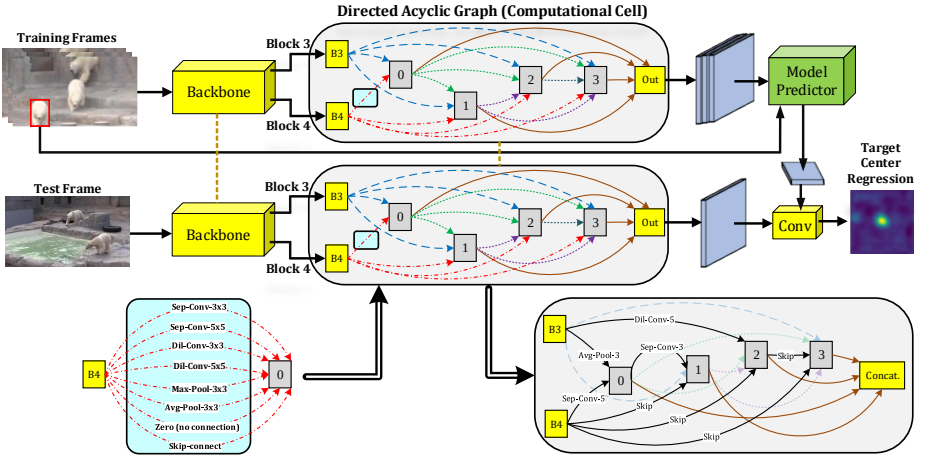

Figure 1: An overview of the proposed CHASE tracker. Cell-level NAS is integrated into the TCR network of the baseline tracker [12] to adapt backbone features to the network objective. First, a computational cell is formed in searching phase in which each edge (dash line) is a mixture of candidate operations (shown as a blue box for one edge), each intermediate node is connected to all the previous nodes, and the output node is the concatenation of intermediate nodes (shown by brown solid lines). The objective of this phase is to find the optimal sub-graph (i.e., the best cell shown at the bottom-right) by jointly optimizing the weights and architecture parameters of the cell. Then, in training phase, the computational cell is replaced by the best cell, and the whole pipeline is trained from scratch. Finally, the network is used in evaluating phase for visual tracking.

Neural architecture search (NAS) has been developed to automatically discover preferable (or ideally optimal) network architecture for a learning task by exploring a wide-reaching space of operation candidates. Generally, NAS methods are classified into the *reinforcement learning* (RL)-based, *evolutionary algorithm* (EA)-based, *Bayesian optimization* (BO)-based, and gradient-based methods, according to their diversified search strategies. Although the first three categories suffer from less efficiency, high time consumption, and extensive computational overhead, the gradient-based methods provide competitive performances & quite efficiency. The well-known *differentiable architecture search* (DARTS) [25] introduces a generic approach that relaxes the search space into the continuous domain and shares the parameters among candidate architectures. Although DARTS has achieved promising results by its gradient-based searches (resulting in the 1st- & 2nd-order versions of DARTS), several works [3, 4, 8, 9, 17, 42] have been proposed to study and address its problems.

Despite the exploitation of NAS in numerous tasks (e.g., classification [8, 25], detection [6, 35], semantic segmentation [24, 31]), almost all the network architectures for visual tracking are based on human-designed heuristics. Very recently, the LightTrack [39] uses evolutionary search to obtain lightweight architectures for resource-limited hardware platforms. Also, it uses single-path uniform sampling and lightweight building blocks to achieve more compact architectures and reduce the computational costs. However, single-path sampling decouple the optimizations of the weights and architecture parameters of the supernet, leading to large-variance to the optimization process and tendency to a non-complex structure [35]. The LightTrack [39] has inherited the limitations of EA-based methods as well as single-pass search approaches. Furthermore, it searches within a limited search space and

stacks the basic blocks to construct the final architecture.

In contrast, the aim of this work is to automatically discover the best architecture block (or cell) that adapts large-scale trained backbone features to the objectives of Siamese tracking networks. Although it modifies DARTS [25] with attractive advantages (e.g., weight-sharing & efficiency), the primary differences include (i) cell-level NAS instead of searching stacked cells together, (ii) integrating cell-level NAS into Siamese framework especially beneficial for visual tracking, (iii) employing operation-level Dropout without hand-crafted constraints used in [4, 5], and (iv) proposing an early-stopping strategy for searching procedure to address the over-fitting problem and multiple retraining from scratch to select the best cell. The proposed approach (CHASE) takes advantage of the 2nd-order DARTS by learning a cell into Siamese tracking networks. This is contrary to prior works (e.g., [4, 5, 8, 9, 17, 25, 42]) searching for multiple stacked cells in CNN/RNN architectures using the simple 1st-order DARTS with lower performance. The CHASE provides a simple, efficient, and generalizable approach considering visual tracking purposes, i.e., high performance & speed. Besides, DARTS-based methods require searching on a small proxy dataset and transferring the architecture blocks to the large-scale target task to address the high GPU memory consumption issues. However, the CHASE performs a cell-level architecture search, which allows directly utilizing a large-scale tracking dataset. Last but not least, this work removes prior heuristics since the proposed early-stopping provides a performance-aware cell derivation strategy during the searching phase. It exploits a hold-out sample set for validating the generalization of the best cell. Thus, it finds the saturated searching point to address the over-fitting problem and the performance gap between the search and evaluation phases [5], and then it can select the best cell without requiring multiple retraining from scratch. Finally, the effectiveness of NAS exploitation and its generalization is validated by employing three versions of DARTS [8, 25] and integrating the proposed approach into two visual trackers [1, 17].

In summary, the main contributions are as follows:

- A novel cell-level differentiable architecture search mechanism is proposed to automate the network design of the tracking module during offline training. It is effectively integrated into Siamese tracking network architectures to directly optimize a cell on a large-scale tracking dataset. Our approach is simple, efficient, and easy to be incorporated into existing trackers for improving performance.
- An early-stopping strategy is proposed to improve the generalization performance of selected cell architecture. This simple yet effective performance-aware cell derivation strategy finds the best cell during the searching phase without requiring inefficient multiple re-training from scratch.
- Extensive experimental evaluations on five widely-used visual tracking benchmarks demonstrate the superior performance of the proposed approach. Moreover, it is practically shown to boost the overall performance when applied to existing baselines.

2 Related Work

2.1 Single Object Tracking

Most recent state-of-the-art visual trackers are based on classic/custom Siamese networks [1, 7, 12, 16, 21, 41] providing a good trade-off between performance & computational complexity. The main ideas include taking powerful backbone features and employing lightweight modules to extract robust target-specific features for visual tracking. For in-

stance, all the SiamRPN++ [21], SiamBAN [7], SiamCAR [16], DiMP [1], SiamAttn [41], and PrDiMP [12] trackers use ResNet-50 [18] as the backbone and adapt the features for visual object tracking using shallow sub-networks. However, these hand-designed sub-networks are biased toward human priors with no guarantees achieving the highest effectiveness. This motivates this work to automatically design these modules by a cell-level search procedure.

2.2 Differentiable NAS

Recently, the gradient-based NAS has shown promising results while searching for a few GPU days. As mentioned before, DARTS [25] is the most popular gradient-based approach introducing the 1st- & 2nd-order approximation-based approaches according to the calculation of architecture gradient, where the 2nd-order one leads to better performance but lower search speed. However, the DARTS suffers from (i) the performance gap between the search & evaluation phases [4, 5], (ii) repeating blocks restriction [3], (iii) performance collapse [9, 17, 42] due to the model over-fitting, (iv) degenerate architectures [42], (v) aggregation of skip connections [4, 5, 8], and (vi) requiring multiple re-training from scratch.

Consequently, several works are presented to address the problems of DARTS. To bridge the gap between the search and evaluation phases, the *progressive DARTS* (PDARTS) [4] gradually increases the network depth assisted by the search space approximation and regularization. The ProxylessNAS [3] proposes learning architectures on large-scale datasets, path-level pruning, and latency regularization loss to address repeating blocks restriction, GPU memory consumption, and hardware limitations. The DARTS+ [17] proposes an early stopping paradigm with hand-crafted constraints to avoid the performance collapse of DARTS due to the model over-fitting in the search phase. To improve the robustness, the RobustDARTS [42] introduces an adaptive regularization and early stopping criterion with the dominant Hessian eigenvalue of validation loss. The DARTS- [9] distinguishes two roles of skip connections (i.e., stabilization of supernet training & candidate operation) by an auxiliary skip connection between every two nodes. Finally, the Fair-DARTS [8] proposes the collaborative competition approach and auxiliary loss to address the aggregation of skip connections & discretization discrepancy problems, respectively.

Most DARTS-based methods (e.g., [4, 5, 8, 9, 17, 42]) employ the 1st-order DARTS to reduce computational complexity, allowing the search procedure on some stacked cells. The 2st-order DARTS fully exploits training & validation information and converging to a better local optimum. This work integrates a modified cell-level 2nd-order DARTS into the Siamese framework to track visual targets. The proposed early-stopping strategy and operation-level Dropout [4, 5] without any constraints are exploited to address the over-fitting problem, test-validation performance gap, and the best cell architecture selection.

3 Proposed Approach: CHASE

The primary motivation is to automatically adapt the robust features extracted from the backbone to the tracking objective by a computational cell (see Fig. 1). Hence, this work exploits a modified version of DARTS [25] that forms an ordered *directed acyclic graph* (DAG) with $\mathcal{N}$ nodes as its computational cell, which is learned through architecture search procedure. The CHASE learns a cell integrated into a Siamese tracking architecture to avoid dramatically affecting the computational complexity & tracking speed. PrDiMP [12] is used as the

baseline to demonstrate the effectiveness of the proposed approach for visual tracking. It includes the *target center regression* (TCR) & *bounding box regression* (BBR) networks, while it predicts the conditional probability density to minimize the *Kullback-Leiber* (KL) divergence between the predictions and label distribution (see [12] for more details). The CHASE tracker replaces additional convolutional blocks after the backbone with a DAG to find the best operations and node connections.

3.1 Cell-Level NAS for Visual Tracking

In this section, DARTS is adapted to a Siamese tracking network to move toward our objectives and critical aspects of visual tracking. In proposed approach, the computational cell has two input nodes and four intermediate nodes. The CHASE fuses multi-level deep features extracted from `Block3` & `Block4` of ResNet-50 [18] in designing the cell, according to their importance for visual tracking [22, 28]. Given a feature map $\mathcal{X}^{(i)}$ at node i, the corresponding latent representation at intermediate node j is computed as $\mathcal{X}^{(j)} = \sum_{i<j} \mathfrak{p}^{(i,j)}(\mathcal{X}^{(i)})$, where $\mathfrak{p}^{(i,j)}$ stands for candidate operations (from a predefined set $\mathcal{P} = \{\mathfrak{p}_1^{(i,j)}, \mathfrak{p}_2^{(i,j)}, ..., \mathfrak{p}_{\mathcal{M}}^{(i,j)}\}$ in the search space) on edge $\zeta^{(i,j)}$. Since the DARTS tends to aggregate skip connections due to the rapid error decay during its optimization [8, 17], the CHASE employs the operation-level Dropout without constraints in [4, 5] with an initial rate τ, which gradually decays during the search procedure. The CHASE does not control the number of skip connections to preserve flexibility in cell design and improve training stability. To relax the problem into a continuous search space, the mixed output for $\zeta^{(i,j)}$ is calculated by

$$\bar{\mathfrak{p}}^{(i,j)}(\mathcal{X}) = \sum_{\mathfrak{p}\in\mathcal{P}} \frac{exp(\alpha_{\mathfrak{p}}^{(i,j)})}{\sum_{\hat{\mathfrak{p}}\in\mathcal{P}} exp(\alpha_{\hat{\mathfrak{p}}}^{(i,j)})} \mathfrak{p}(\mathcal{X}), \tag{1}$$

in which $\alpha_{\mathfrak{p}}^{(i,j)}$ is the operation mixing weight associated with the operation $\mathfrak{p}$ between nodes i and j. By doing so, the cell architecture search converts into the learning of parameters $\alpha = \{\alpha_1^{(i,j)}, \alpha_2^{(i,j)}, ..., \alpha_{\mathcal{M}}^{(i,j)}\}$. To jointly learn network parameters ($\mathcal{W}$) and architecture parameters (α), the *gradient descent* (GD) algorithm is used to minimize the training ($\mathcal{L}_{tr}$) and validation losses ($\mathcal{L}_{val}$) by performing the bi-level optimization problem

$$\min_{\alpha} \ \mathcal{L}_{val}(\mathcal{W}^*(\alpha), \alpha) \tag{2}$$

$$\text{s.t.} \ \ \mathcal{W}^*(\alpha) = \underset{\mathcal{W}}{\operatorname{argmin}} \mathcal{L}_{tr}(\mathcal{W}, \alpha). \tag{3}$$

To avoid expensive inner optimization, the DARTS reduces the evaluation of architecture gradient by applying the finite difference approximation. By doing so, the 2nd-order approximation of DARTS requires two forward passes for $\mathcal{W}$ and two backward passes for α, contrary to the 1st-order DARTS requiring one forward pass for each one (see [25] for more details).

The 1st-order DARTS provides the ability to search an architecture by stacking multiple cells according to its simplicity and low complexity, e.g., [4, 5, 8, 9, 17, 42]. Although differentiable NAS aims at minimizing the validation loss to find optimal architectures, the 1st-order DARTS cannot guarantee that the validation loss is sufficiently small due to ignoring the optimization on fully-trained weights $\mathcal{W}^*(\alpha)$. The 2nd-order DARTS embeds the training loss in updating architecture parameters. Hence, it achieves more stability and higher performance than the 1st-order DARTS by fully exploiting training & validation information and converging to a better local optimum. However, it increases the computational

complexity not efficient for optimizing stacked cells. The CHASE enjoys the modified 2nd-order DARTS according to learning one cell that adapts large-scale trained backbone features to the tracking objectives. Moreover, the DARTS [25] suffers from some problems as i) deriving the best discrete architecture with the best validation performance by re-training top-k architectures ($k = 4$) from scratch, and ii) the performance collapse and over-fitting problems on the validation set, resulting in poor generalization on test datasets. To address these challenges, the proposed CHASE focuses on cell-level search and proposes an early stopping strategy to address the over-fitting problem and multiple re-training from scratch.

3.2 Early Stopping

To alleviate the test-validation gap of DARTS, prior works (e.g., [17, 42]) impose strong early stopping priors or extra computing costs. However, these methods run several times and re-train each best architecture from scratch to select the final one. This work performs a performance-aware cell derivation by the proposed early stopping strategy to address these limitations simultaneously. In particular, generic visual tracking seeks to learn target models generalizable to various appearance changes and real-world challenging scenarios. Hence, the proposed strategy introduces a hold-out sample set represented for generalization validation. Note that the CHASE never uses test sets for this purpose. While the CHASE respectively optimizes $\mathcal{W}$ and α on the training and validation sets, it calculates the hold-out loss ($\mathcal{L}_{ho}$) of mixture operations. Then, it derives the best cell architecture at the minimum hold-out loss on the hold-out set by $\mathfrak{p}_o^{(i,j)} = \text{argmax}_{\mathfrak{p}\in\mathcal{P}} \alpha_{\mathfrak{p}}^{(i,j)}$. This search-stage cell selection originates from the reduced discrepancies between the continuous cell encoding and the derived discrete cell due to the searching one cell using the proposed modified 2nd-order DARTS, resulting in no several re-training requirements from scratch. That is, the CHASE finds the best cell during the searching phase and then trains it from scratch once.

4 Empirical Experiments

Herein, the implementation details of the proposed approach, ablation analysis, and tracking results of the best cell architecture on benchmark datasets are reported. Also, codes & experimental results are publicly available on github.com/VisualTrackingVLL.

4.1 Implementation Details

The backbone consists of ResNet-50 architecture [18] initialized with the pre-trained ImageNet [32] weights. The offline experiments comprise the searching and training phases. The proposed CHASE tracker is implemented in PyTorch and runs 23 *fps* on a single Nvidia Tesla V100 GPU with 16GB RAM. Except for the following details, the rest of the hyper-parameters are set to the ones in [17]. The test sets are never utilized in searching or training phases.

4.1.1 Searching Phase

In this phase, the cell architecture is searched by the modified 2nd-order DARTS. The cell includes 14 edges and 7 nodes (2 input, 4 intermediate, and 1 output), which the output node is obtained by depthwise concatenation of intermediate nodes. The standard DARTS

search space is employed to exploit the maximum number of nodes & edges allowing in a cell, which provides the highest flexibility in cell design. The candidate operations include 3×3 & 5×5 separable convolutions, 3×3 & 5×5 dilated convolutions, 3×3 max pooling, 3×3 average pooling, zero (no connection), and skip connection (i.e., $\mathcal{M} = 8$). The CHASE applies operation-level Dropout, which its rate starts from $\tau = 0.6$ and gradually decayed to the last epoch. In contrast to [4, 5], the CHASE fairly explores all operations, considering the importance of skip-connections on the evaluation accuracy and architecture stability.

The training set of the TrackingNet dataset [30] is divided into two subsets for optimizing the weights of network ($\mathcal{W}$) & encoding weights of architecture (α) on the training ($\mathcal{L}_{tr}$) & validation ($\mathcal{L}_{val}$) sets, respectively. Besides, the training sets of GOT-10k [19] and LaSOT [14] datasets are used as the hold-out set ($\mathcal{L}_{ho}$) to specify the best architecture among three runs (with different random seeds) and select the final cell architecture based on their performance. Based on the training tricks of NAS in [37], the backbone and BBR parameters are frozen during architecture search, while the architecture parameters are started to optimize after 10 epochs. It is more critical for the proposed approach to calculate reliable 2nd-order gradients of architecture parameters built on 1st-order ones of network weights. The proposed CHASE provides better initialization of candidate operations directly impacting the optimization procedure of architecture parameters. Thus, it provides fair competition between weight-free operations with other ones and helps effective learning of architecture parameters, leading to performance improvement, acceleration, and avoiding getting stuck into bad local optima. The network is trained for at most 70 epochs with a batch size of 10, similar to the baseline [12]. However, the proposed approach stops the training procedure based on the proposed early-stopping strategy (epoch 41 for CHASE). The Adam optimizer [20] is used to learn network and architecture parameters. The initial learning rate is 0.001 for optimizing $\mathcal{W}$ with the cosine annealing scheduler. The maximum iteration numbers are 15K, 15K, and 5K for training, validation, and early-stopping procedures. The search phase takes about 41 (18) hours for the second (first) order DARTS method using the TrackingNet dataset on a Nvidia Tesla V100 GPU with 16GB RAM.

4.1.2 Training Phase

In contrast to prior works (e.g., [4, 5, 9, 17, 25, 42]), the CHASE just trains the best model selected in searching phase from scratch. In this phase, computational cell is replaced by the best cell architecture, and the whole network (including backbone, TCR, and BBR) is jointly trained from scratch for 70 epochs. The TCR and BBR layers are initialized with random weights ignoring the weights during the searching phase. For the training phase, the training sets of LaSOT [14], TrackingNet [30], GOT-10k [19], and COCO [23] datasets are used, similar to the baseline [12]. Also, other hyper-parameters are set as in the baseline tracker [12].

4.1.3 Evaluating Phase

After offline training phases, the proposed CHASE tracker is evaluated on test splits of generic and aerial visual tracking datasets, namely GOT-10k [19], TrackingNet [30], LaSOT [14], UAV-123 [29], and VisDrone-2019-test-dev [13]. In the online phase, all procedures and settings are the same as [12].

4.2 Ablation Analysis

In this section, a systematic ablation analysis on the GOT-10k dataset [19] is conducted to validate the effectiveness of various search spaces and methods. It includes the cells derived by the 1) 1st-order DARTS (CHASE-D1), 2) Fair-DARTS [8] (CHASE-FD), and 3) proposed approach (CHASE-PrDiMP or CHASE). Besides, the CHASE is integrated into the DiMP tracker [1] (CHASE-DiMP), demonstrating the generalization of the proposed approach for visual tracking. Furthermore, three versions of the proposed approach are investigated, including the CHASE with 1) fully segregated datasets in searching & training phases (CHASE-S/T), 2) a search space consisting of two intermediate nodes (CHASE-2N), and 3) a search space without weightless candidate operations (CHASE-WO). The comparison results are reported in Table 1 regarding the derived cells shown in Fig. 2.

Accordingly, the CHASE-D1 derives a cell dominated by weight-free operations (i.e., skip and pooling operations), and there is no connection between intermediate nodes resulting in a shallow architecture. The CHASE-FD employs the Fair-DARTS [8], which utilizes the Sigmoid activation function and an auxiliary loss to address exclusive competition of skip-connections and discretization discrepancy. Nonetheless, the CHASE outperforms the CHASE-D1 & CHASE-FD up to 3.6% and 2.1% in terms of *average overlap* (AO) metric, respectively. Conventional DARTS-based methods (with stacked cell networks for image classification) search a network architecture on a small proxy dataset (e.g., CIFAR-10) and then transfer it to a large-scale target dataset (e.g., ImageNet) to alleviate high memory consumption [4]. However, the proposed approach can enjoy searching on the large-scale TrackingNet dataset by its cell-level search. Hence, the CHASE uses the large-scale TrackingNet dataset in both searching & training phases outperforming the CHASE-S/T up to 1.4% in terms of AO metric. Except for CHASE-S/T, all CHASE-versions have been searched and trained on similar datasets mentioned in Sec. 4.1.1 and Sec. 4.1.2, respectively.

While the CHASE employs the standard DARTS search space to have more design flexibility via the maximum number of nodes & edges allowing in a cell, the CHASE-2N

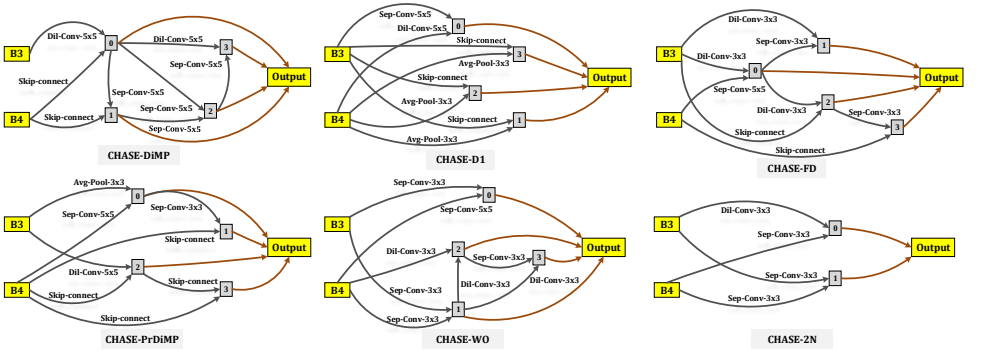

Figure 2: Best cell architectures derived by CHASE-DiMP (modified 2nd-order DARTS), CHASE-D1 (1st-order DARTS), CHASE-FD (Fair-DARTS), CHASE-PrDiMP (modified 2nd-order DARTS), CHASE-WO (modified 2nd-order DARTS without weightless operations), and CHASE-2N (modified 2nd-order DARTS with two intermediate nodes). B3 and B4 are the input latent representations (from `Block3` & `Block4` of Resnet50 [18], respectively). Also, 0 , 1 , 2 , 3 are the intermediate nodes, and the output is the depthwise concatenation of intermediate nodes.

Table 1: Ablation analysis of CHASE on GOT-10k dataset [19].

Metric	DiMP [1]	**CHASE-DiMP**	PrDiMP [12]	CHASE-D1	CHASE-FD	**CHASE**	CHASE-2N	CHASE-WO	CHASE-S/T
$SR_{0.75}$ (↑)	49.2	**51.1**	54.3	54.8	56.1	**56.5**	51.4	45.9	56.1
$SR_{0.5}$ (↑)	71.7	**75.3**	73.8	76.7	76.8	**78.8**	76.5	71.5	76.3
AO (↑)	61.1	**63.6**	63.4	64.9	65.6	**67.0**	64.2	60.7	65.6

and CHASE-WO represent search spaces with limited node numbers (i.e., two intermediate nodes) and removed weightless candidate operations (i.e., pooling, zero, & skip connect), respectively. According to the results, the CHASE has improved the performance of CHASE-2N & CHASE-WO up to 2.8% & 6.3% in terms of AO metric, respectively. These results demonstrate prior heuristics and limited search space dramatically affect architecture design and tracking performance. For instance, the intuitive reason in the case of CHASE-WO is that removing weightless operations (particularly skip-connections) has been led to instability in cell design and accuracy degradation. Besides, the node restriction results in shallow cell architecture and limited performance improvement. The computational cells derived by the CHASE-PrDiMP confirm selecting various operations regarding objective function, increasing the depth as necessary, and preventing over-fitting and performance collapse problems. Finally, the proposed approach is integrated into the DiMP tracker [1] minimizing an L^2-based discriminative learning loss to train its network to investigate the generalization to different objective functions. The proposed approach outperforms the DiMP tracker [1] up to 2.5% in terms of the AO and up to 3.6% in terms of *success rate* (SR) at the overlap threshold of 0.5. At last, the best-performing tracker, CHASE, is selected to be compared with recent trackers in the next section.

4.3 State-of-the-art Comparison

In this section, the state-of-the-art evaluations are performed on five large-scale visual tracking benchmarks (refer to Sec. 4.1.3) and the proposed CHASE tracker is compared with various state-of-the-art visual trackers, namely ECO [10], SiamMask [36], DaSiamRPN [45], SiamRPN++ [21], ATOM [11], DCFST [44], COMET [27], SiamFC++ [38], DiMP-50 [1], PrDiMP-50 [12], KYS [2], SiamAttn [41], MAML [15], ROAM++ [40], SiamCAR [16], SiamBAN [7], D3S [26], Ocean [43], and LightTrack [39].

GOT-10k [19]: This large high-diversity dataset includes over 10K videos as the training set and 180 videos for evaluation without publicly available ground-truth. Notably, the target classes for evaluation do not overlap with training ones. Hence, this dataset is usually used for studying the transferability of proposed approaches for tracking unseen targets. Therefore, the proposed CHASE uses its training set as one of the hold-out sets to early-stop the cell searching phase. The comparison results presented in Table 2 show that the CHASE outperforms the baseline up to 3.6%, 5%, and 2.2% in terms of AO and SR at overlap thresholds of 0.5 and 0.75, respectively. Besides, the CHASE has achieved better results (4.7% in AO, 6.2% in $SR_{0.5}$) compared with the LightTrack [39].

LaSOT [14]: LaSOT is a long-term and challenging tracking benchmark consisting of 1400 videos and 3.5M frames, with 2500 frames per video on average. The test set contains 280 videos and 690K frames with target disappear/reappear scenarios. Thus, this dataset appropriately indicates the robustness of short-term trackers in real-world situations. For this reason, the proposed tracker uses its training set as the second dataset of hold-out set in the searching phase. As shown in Table 2, the CHASE improves the baseline results [12] by a margin of 1.9%, 2.3%, and 2.1% in terms of *area under curve* (AUC), normalized precision, and precision, respectively.

Table 2: State-of-the-art comparison results on GOT-10k [19], LaSOT [14], TrackingNet [30], UAV-123 [29], VisDrone-2019-test-dev [43] datasets.

Trackers	GOT-10k			LaSOT			TrackingNet			UAV-123		VisDrone-2019-test-dev	
	AO (↑)	$SR_{0.5}$ (↑)	$SR_{0.75}$ (↑)	AUC (↑)	Norm. Prec. (↑)	Prec. (↑)	AUC (↑)	Norm. Prec. (↑)	Prec. (↑)	$SR_{0.5}$ (↑)	Prec. (↑)	AUC (↑)	Prec. (↑)
CHASE	**67.0**	**78.8**	**56.5**	**61.7**	**71.1**	**62.9**	**76.8**	**82.5**	**71.8**	**83.9**	**88.2**	**61.7**	82.0
LightTrack [24]	62.3	72.6	-	-	-	56.1	73.3	78.9	70.8	-	-	-	-
PrDiMP-50 [8]	63.4	**73.8**	**54.3**	**59.8**	**68.8**	**60.8**	**75.8**	81.6	70.4	**82.7**	**87.4**	59.8	79.7
Ocean [44]	61.1	72.1	47.3	56.0	65.1	56.6	-	-	-	-	-	59.4	82.3
D3S [28]	59.7	67.6	46.2	-	-	-	72.8	76.8	66.4	-	-	-	-
ROAM++ [40]	46.5	53.2	23.6	44.7	-	44.5	67.0	75.4	62.3	-	-	-	-
SiamAttn [41]	-	-	-	56.0	64.8	-	75.2	**81.7**	-	79.4	84.5	-	-
KYS [3]	63.6	75.1	51.5	55.4	63.3	-	74.0	80.0	68.8	-	-	-	-
DiMP-50 [3]	61.1	71.7	49.2	56.9	65.0	56.7	74.0	80.1	68.7	80.4	85.5	60.8	80.5
SiamCAR [21]	56.9	67.0	41.5	50.7	60.0	51.0	-	-	-	77.3	81.3	-	-
SiamBAN [7]	-	-	-	51.4	59.8	52.1	-	-	-	77.4	83.3	-	-
MAML [23]	-	-	-	52.3	-	-	75.7	82.2	**72.5**	-	-	-	-
ATOM [23]	55.6	63.4	40.2	51.5	57.6	50.5	70.3	77.1	64.8	78.9	85.6	57.1	76.7
SiamRPN++ [20]	51.8	61.8	32.5	49.6	56.9	-	73.3	80.0	69.4	78.8	84.0	59.9	79.1
DCFST [42]	**63.8**	75.3	49.8	-	-	-	75.2	80.9	70.0	-	-	-	-
COMET [27]	59.6	70.6	44.9	54.2	-	-	-	-	-	79.4	86.1	**64.5**	**83.9**
SiamFC++ [39]	59.5	69.5	47.9	54.4	62.3	54.7	75.4	80.0	70.5	-	-	-	-
SiamMask [38]	51.4	58.7	36.6	-	-	-	72.5	77.8	66.4	-	-	58.1	79.4
DaSiamRPN [45]	-	-	-	-	-	-	63.8	73.3	-	72.6	78.1	-	-
ECO [23]	31.6	30.9	11.1	32.4	33.8	30.1	55.4	61.8	49.2	63.1	74.1	55.9	**82.6**

TrackingNet [30]: TrackingNet is a challenging in-the-wild tracking dataset consisting of 27 classes of targets from YouTube videos. This dataset contains more than 30K videos and 14.4M frames, including 500 videos for testing which the ground-truths are not publicly available. From Table 2, the MAML tracker [34] has close results (better in precision metric) compared with the proposed tracker since it employs a modern object detector (i.e., FCOS [33]) and online domain adaptation to enhance discriminating target from non-target regions. However, the proposed CHASE tracker has achieved better results in terms of AUC and normalized precision, and it has improved the baseline results by a margin of 1% in AUC and 1.4% in precision metric.

UAV-123 [29]: UAV-123 is an challenging aerial-view tracking dataset consisting of 123 videos, 113K frames, and 9 classes of targets captured from a low-altitude perspective. According to the results in Table 2, the proposed CHASE tracker outperforms the state-of-the-art visual trackers but also the baseline tracker [12] up to 1.2% and 0.8% in terms of success and precision rate metrics.

VisDrone-2019-test-dev [43]: VisDrone-2019 also aims to track visual targets captured from aerial-view. It includes 35 test videos (112K frames) from challenging scenarios such as abrupt camera motion, tiny targets, fast view-point change, and day/night conditions. Compared with the baseline [12], the results of the CHASE tracker have improved up to 1.9% in AUC and 2.3% in precision rate. The COMET [27] has obtained the best results employing the training set of VisDrone for its offline training and accurately designed modules for small object tracking.

5 Conclusion

A novel cell-level differentiable architecture search mechanism is proposed. To address the inherent limitations of differentiable architecture search, we exploit the second-order DARTS by operation-level dropout without any post-processing and introduce early stopping to mitigate the over-fitting and performance collapse issues. Our approach is simple, efficient, and easy to be integrated into existing visual trackers. Extensive experiments demonstrate the effectiveness of the proposed approach, as well as noticeable performance improvement when working with different existing trackers.

Acknowledgement: This research was partly supported by the NSERC Discovery Grant (No. RGPIN-2019-04575) and the UAHJIC Grants.

References

[1] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proc. ICCV*, pages 6181–6190, 2019.

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Timofte Radu. Know your surroundings: Exploiting scene information for object tracking. In *Proc. ECCV*, 2020.

[3] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *Proc. ICLR*, 2019.

[4] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proc. ICCV*, pages 1294–1303, 2019.

[5] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive DARTS: Bridging the optimization gap for nas in the wild. *Int J Comput Vis*, 129:638–655, 2021.

[6] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. DetNAS: Backbone search for object detection. In *Proc. NeurIPS*, 2019.

[7] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *Proc. IEEE CVPR*, 2020.

[8] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair DARTS: Eliminating unfair advantages in differentiable architecture search. In *Proc. ECCV*, 2020.

[9] Xiangxiang Chu, Xiaoxing Wang, Bo Zhang, Shun Lu, Xiaolin Wei, and Junchi Yan. DARTS-: Robustly stepping out of performance collapse without indicators. In *Proc. ICLR*, 2021.

[10] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *Proc. IEEE CVPR*, pages 6931–6939, 2017.

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proc. IEEE CVPR*, 2019.

[12] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proc. IEEE CVPR*, 2020.

[13] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and et al. VisDrone-SOT2019: The Vision Meets Drone Single Object Tracking Challenge Results. In *Proc. ICCVW*, 2019.

[14] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proc. IEEE CVPR*, 2019.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. ICML*, pages 1126–1135, 2017.

[16] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *Proc. IEEE CVPR*, 2020.

[17] Liang Hanwen, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. DARTS+: Improved differentiable architecture search with early stopping, 2020. URL http://arXiv:1909.06035.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, 2016.

[19] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1562–1577, 2021.

[20] Diederik P. Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. In *Proc. ICLR*, 2014.

[21] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *Proc. IEEE CVPR*, 2019.

[22] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognit.*, 76:323–338, 2018.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, pages 740–755, 2014.

[24] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L. Yuille, and Li Fei-Fei. Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation. In *Proc. IEEE CVPR*, 2019.

[25] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *Proc. ICLR*, 2019.

[26] Alan Lukezic, Jiri Matas, and Matej Kristan. D3S - A discriminative single shot segmentation tracker. In *Proc. IEEE CVPR*, 2020.

[27] Seyed Mojtaba Marvasti-Zadeh, Javad Khaghani, Hossein Ghanei-Yakhdan, Shohreh Kasaei, and Li Cheng. COMET: Context-aware IoU-guided network for small object tracking. In *Proc. ACCV*, 2020.

[28] Seyed Mojtaba Marvasti-Zadeh, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Deep learning for visual tracking: A comprehensive survey. *IEEE Trans. Intell Transp Syst*, pages 1–26, 2021. doi: 10.1109/TITS.2020.3046478.

[29] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *Proc. ECCV*, pages 445–461, 2016.

[30] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *Proc. ECCV*, pages 310–327, 2018.

[31] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *Proc. IEEE CVPR*, 2019.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3): 211–252, 2015.

[33] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proc. ICCV*, 2019.

[34] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *Proc. IEEE CVPR*, 2020.

[35] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. NAS-FCOS: Fast neural architecture search for object detection. In *Proc. IEEE CVPR*, 2020.

[36] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H.S. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proc. IEEE CVPR*, 2019.

[37] Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Auto-FPN: Automatic network architecture adaptation for object detection beyond classification. In *Proc. ICCV*, 2019.

[38] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proc. AAAI*, pages 12549–12556, 2020.

[39] Bin Yan, Houwen Peng, Kan Wu, Dong Wang, Jianlong Fu, and Huchuan Lu. LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search, 2021. URL http://arXiv:2104.14545.

[40] Tianyu Yang, Pengfei Xu, Runbo Hu, Hua Chai, and Antoni B. Chan. ROAM: Recurrently optimizing tracking model. In *Proc. IEEE CVPR*, 2020.

[41] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *Proc. IEEE CVPR*, 2020.

[42] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. In *Proc. ICLR*, 2020.

[43] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *Proc. ECCV*, 2020.

[44] Linyu Zheng, Ming Tang, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Learning feature embeddings for discriminant model based tracking. In *Proc. ECCV*, 2020.

[45] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware Siamese networks for visual object tracking. In *Proc. ECCV*, volume 11213 LNCS, pages 103–119, 2018.