Serial Over Parallel: Learning Continual Unification for Multi-Modal Visual Object Tracking and Benchmarking

Supplementary Material

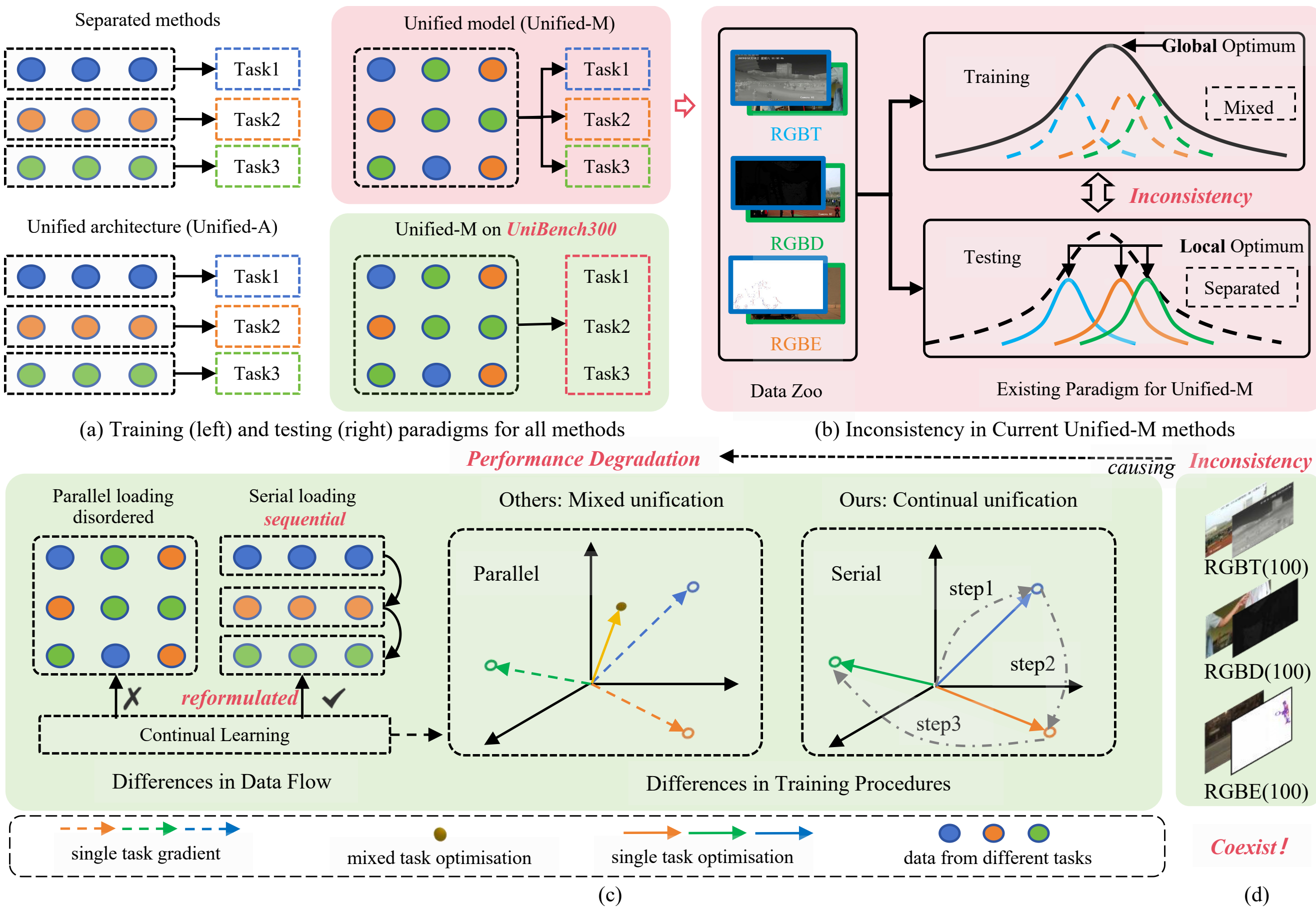

Figure 1: (a) Training (left) and testing (right) paradigms for all methods; (b) Inconsistency between the current training and testing paradigms (global vs. local) in unified methods (Unified-M) leads to performance degradation on separated benchmarks. To address these issues, (d) UniBench300 is proposed as the first unified benchmark to bridge the inconsistency, and (c) the unification process is reformulated as a serial one, facilitating the injection of CL to mitigate performance degradation.

Abstract

This is the supplementary material of ACMMM 2025 paper entitled "Serial Over Parallel: Learning Continual Unification for Multi-Modal Visual Object Tracking and Benchmarking". In this file, the following contents are included:

- Section.1: A brief review of this work is involved to make this file more self-contained.
- Section.2: Detailed efficiency analysis of the proposed SymTrack
- Section.3: Results on all the previous tasks.
- Section.4: Qualitative analysis of the superiority of CL in the embedding space.

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754879

• Section.5: Pseudo code for the proposed multi-step training strategy.
• Section.6: Evaluation metrics for UniBench300.
• Section.7: Insights for the sequence in continual unification.

ACM Reference Format:
Supplementary Material. 2025. Serial Over Parallel: Learning Continual Unification for Multi-Modal Visual Object Tracking and Benchmarking. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3746027.3754879

1 Brief Introduction

Figure 1 presents a brief introduction of this work. As shown in Figure 1(a), existing unified methods fall into two categories: methods with unified architectures (Unified-A) [2–4, 7, 9, 16, 17] and those with unified models (Unified-M) [10, 14]. While Unified-A methods employ a same structure across different tasks, they still require task-specific adaptations, leading to multiple independently trained models rather than a single unified model that integrates the advantages of all data modalities. Therefore, further discussions primarily focus on the second category (Unified-M).

Motivation: To achieve a unified model, data preparation is the first and most important step. As shown in Figure 1(b), existing methods with unified models directly mix all types of data into a unified data zoo [10, 14]. During training, different multi-modal data are loaded in parallel, infusing multi-modal knowledge into the unified model with the goal of finding a global optimum on the joint distribution . However, this contradicts the testing phase, where unified models are evaluated on separated benchmarks, in the same way with methods trained separately [1, 8, 11]. It means local optimum is preferred and this preference introduces an inconsistency between training and testing, ultimately leading to performance degradation.

To address these issues, our work makes two crucial advancements: ❶ As shown in Figure 1(d), to resolve the inconsistency issue, UniBench300 is introduced as the first unified benchmark for MMVOT. It comprises 300 video sequences, including 100 RGBT sequences, 100 RGBD sequences, and 100 RGBE sequences, and 368.1K frames in total. By forming a joint distribution of multi-modal data, UniBench300 aligns the training and testing paradigms, thus bridging the inconsistency. Additionally, UniBench300 enhances evaluation convenience and efficiency by reducing inference time by 27% while requiring only a single evaluation pass (three times before). ❷ From a data perspective, unification can follow either a parallel or serial approach. While existing works [10, 14] employ parallel unification, they suffer from performance degradation, necessitating an exploration of serial unification. As depicted in Figure1(c), serial unification progressively integrates new tasks, specifying performance degradation as knowledge forgetting of previous tasks, which is a core topic in the realm of continual learning (CL) [12]. Based on this, reformulating the unification process into a serial one enables the natural incorporation of CL techniques into the unification of MMVOT tasks. It benefits unified models with better efficacy on both previous and new tasks. As demonstrated in Figure 1, extensive experiments on two baselines and four benchmarks validate the superiority of CL in stabilising performance across tasks.

2 Efficiency Analysis

Table 1: Detailed Efficiency Analysis of ViPT and SymTrack

	ViPT	SymTrack
Real-Time	25 FPS	
LasHeR	66 FPS	56 FPS
VisEvent	95 FPS	85 FPS
DepthTrack	70 FPS	61 FPS
UniBench300	73 FPS	65 FPS
FLOPS	29 G	66 G
Parameters	93 M	138 M

Table 1 presents the efficiency analysis of the involved methods, ViPT* [17] and SymTrack. As shown in the table, ViPT* (SymTrack) achieves 66 (56), 95 (85), 70 (61), and 73 (65) frames per second (FPS) on LasHeR [6], VisEvent [13], DepthTrack [15], and the proposed UniBench300, respectively. Both methods operate well above the real-time threshold of 25 FPS, making them viable options for practical deployment. Additionally, Table 1 reports the computational complexity in terms of floating-point operations (FLOPs) and network parameters. ViPT* and SymTrack require 29G and 66G FLOPs, and contain 93M and 138M parameters, respectively.

Notably, different from some works that present solely the efficiency of the core function, we report the efficiency of the entire tracking process, including pre- and post-processes, which are executed on the CPU. Moreover, the image size of RGBE data is much smaller than that of RGBT and RGBD data, which leads to a better balance of disk I/O, CPU and GPU utilisation, resulting in much higher efficiency on RGBE task. Besides, the different length of RGBT/RGBD sequences also contributes to the slight fluctuation in efficiency.

3 Detailed Performance Analysis

Different from reporting performance on a single primary task, Table 2 provides a fine-grained analysis of the proposed continual unification process enhanced by CL on more combinations of involved tasks. In this table, "mixed" denotes the original parallel unification paradigm, where data from all tasks are disorderly mixed and loaded. "CL" indicates that the variants are trained under the continual unification paradigm. T, D, and E represent the RGBT, RGBD, and RGBE tasks, respectively. It is evident that using the original paradigm "mixed" leads to performance degradation after unification on all tasks and all training steps. In contrast, with the proposed "CL" serial paradigm, performance across all tasks and steps is consistently maintained—showing only minor fluctuations. Furthermore, under the same training and testing data conditions, almost all variants trained with the "CL" paradigm outperform those trained with "mixed", significantly demonstrating the effectiveness of continual learning in stabilising the unification process.

4 Pseudo Code of Continual Unification

Step 1: Initialising the network *randomly* and training the model with LasHeR. In this way, the trained model SymTrack-t.pth can be evaluated on the testing split of LasHeR.

Table 2: Detailed Quantitative Comparisons between parallel and continual unification.

Variants	Train	Test										
	T	T	D	E	D	T	D	E	E	T	D	E
ViPT*	T	0.519(-0.0)			D		0.598(-0.0)		E			0.591(-0.0)
+mixed	T+D	0.510(-0.9)	0.582(-0.0)		D+E		0.585(-1.3)	0.589(-0.0)	E+T	0.513(-0.0)		0.582(-0.9)
	T+D+E	0.494(-1.6)	0.573(-1.1)	0.579(-0.0)	D+E+T	0.494(-0.0)	0.573(-1.2)	0.579(-1.0)	E+T+D	0.494(-1.9)	0.573(-0.0)	0.579(-0.3)
+CL	T+D	0.525(+0.6)	0.584(-0.0)		D+E		0.597(-0.1)	0.583(-0.0)	E+T	0.508(-0.0)		0.588(-0.3)
	T+D+E	0.527(+0.2)	0.599(+1.5)	0.582(-0.0)	D+E+T	0.499(-0.0)	0.596(-0.1)	0.589(+0.6)	E+T+D	0.510(+0.2)	0.573(-0.0)	0.588(+0.0)
Δ	T+D	+1.5	+0.2		D+E		+1.2	-0.6	E+T	-0.5		+0.6
	T+D+E	+3.3	+2.5	+0.3	D+E+T	+0.5	+2.3	+1.0	E+T+D	+1.6	+0.0	+0.9

Table 3: Pseudo Code of Continual Unification.

Step	Pseudo code of Continual Unification Procedure
0	Prepare - Datasets: LasHeR, DepthTrack, VisEvent - Method: SymTrack - Determine the sequence (RGBT-RGBD-RGBE)
1	Start Training - Model: Random initialised - Training Set(s): LasHeR - Testing Set(s): LasHeR - Saved Model: SymTrack-t.pth
2	Continual Aggregation - Model: Initialised by SymTrack-t.pth - Training Set(s): LasHeR, DepthTrack - Testing Set(s): LasHeR, DepthTrack - Saved Model: SymTrack-td.pth
3	Continual Aggregation - Model: Initialised by SymTrack-td.pth - Training Set(s): LasHeR, DepthTrack, VisEvent - Testing Set(s): LasHeR, DepthTrack, VisEvent - Saved Model: SymTrack-tde.pth
4	End Training

Step 2: Initialising the network by *SymTrack-t.pth* and training the model with LasHeR and DepthTrack. In this way, the trained model SymTrack-td.pth can be evaluated on the testing splits of LasHeR and DepthTrack.

Step 3: Initialising the network by *SymTrack-td.pth* and training the model with LasHeR, DepthTrack, and VisEvent. In this way, the trained model SymTrack-tdd.pth can be evaluated on the testing splits of LasHeR, DepthTrack, and VisEvent.

To intuitively present the proposed continual unification process, Table 3 provides the pseudo-code for the training procedure. In general, continual unification consists of five key steps:

Step 0: Prepare the datasets (LasHeR [6], DepthTrack [15], and VisEvent [13]), select the implemented methods (SymTrack), and determine a sequence for progressively integrating multi-task knowledge (e.g., T → D → E).

Step 1: *Randomly* initialise the network and train the model on LasHeR. The trained model, SymTrack-t.pth, can then be evaluated on the test split of LasHeR.

Step 2: Initialise the network using *SymTrack-t.pth* and train the model on LasHeR and DepthTrack. The resulting model, SymTrack-td.pth, can be evaluated on the test splits of LasHeR and DepthTrack.

Step 3: Initialise the network using *SymTrack-td.pth* and train the model on LasHeR, DepthTrack, and VisEvent. The final model, SymTrack-tde.pth, can then be evaluated on the test splits of LasHeR, DepthTrack, and VisEvent.

Step 4: Training procedure completes.

5 Quantitative Analysis of CL

Figure 2 compares models trained under the original parallel paradigm and the proposed continual unification process on UniBench300. In this figure, the classification maps of the second frame are visualised to ensure both methods receive identical input search regions. As a result, both maps appear relatively clean. However, it is still evident that the variant trained with continual learning contains less noise than the one trained with disordered multi-task data. This distinction is especially clear in the third example, where SymTrack+mixed shows a strong response to a distractor, whereas SymTrack+CL shows minimal response, highlighting the advantage of continual unification.

6 Evaluation Metrics for UniBench300

UniBench300 adopts precision rate (PR) and success rate (SR) as evaluation metrics, aligning with those used in established benchmarks such as VisEvent [5] and LasHeR [6]. PR quantifies the percentage of frames where the centre distance between the predicted and ground truth bounding boxes falls below a predefined threshold. SR measures the proportion of frames where the predicted bounding box maintains an overlap with the ground truth. The mathematical definitions are as follows:

$$\begin{aligned} \mathrm{SR} &= \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{t}\sum_{i=1}^{t}\mathrm{IoU}(\boldsymbol{g}_{j,i}, \boldsymbol{p}_{j,i}) > \mathrm{th_s}\right) \\ \mathrm{PR} &= \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{t}\sum_{i=1}^{t}\mathrm{Dis}(\boldsymbol{g}_{j,i,c}, \boldsymbol{p}_{j,i,c}) > \mathrm{th_p}\right) \end{aligned} \tag{1}$$

where the IoU between the ground truth bounding box $\boldsymbol{g}_{j,i}$ and predicted bounding box $\boldsymbol{p}_{j,i}$ is computed for evaluation, along with the ℓ_2 distance (Dis) between the centres of these bounding boxes,

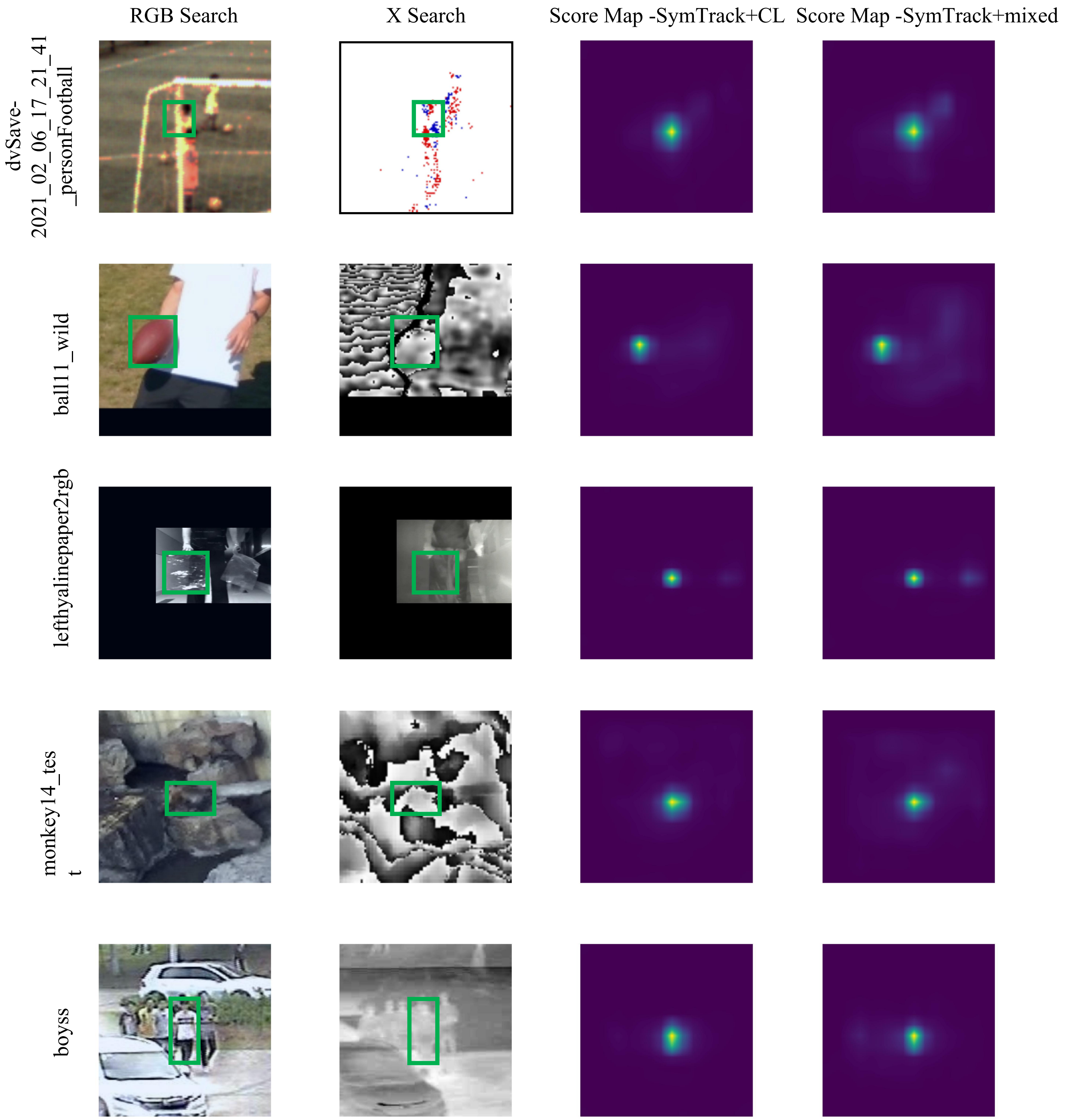

Figure 2: Qualitative analysis of the superiority of CL in the embedding space. Better viewed after zoomed in.

$\boldsymbol{g}_{j,i,c}$ and $\boldsymbol{p}_{j,i,c}$. Subscript j denotes the j-th sequence and i represents the i-th frame. c indicates the centre of the bounding box. t and m refer to the number of frames in a sequence and the number of sequences contained in the entire benchmark, respectively. This means IoU and the centre distance are first averaged over all frames within each sequence and then across all sequences. The threshold $\mathrm{th_s}$ and $\mathrm{th_p}$ are utilised for calculating SR and PR, respectively. To ensure a comprehensive evaluation, multiple thresholds are applied

Table 4: Results of variants in different sequence on UniBench300.

Variant	SR
SymTrack+CL+TDE	0.395
SymTrack+CL+TED	0.394
SymTrack+CL+ETD	0.391
SymTrack+CL+EDT	0.386
SymTrack+CL+DTE	0.386
SymTrack+CL+DET	0.383

and the results under each threshold are recorded. The final score is reported as the area under curve (AUC). Notably, only frames where the object remains visible are included in evaluation.

7 Insights for the Sequence in Continual Unification

In this work, CL is introduced to prevent the knowledge forgetting of previous tasks. It means, in the last step of the training process, the integration of the last task still falls the dilemma with the original unification paradigm. Instead of introducing further techniques to solve this issue, we find another approach to relieve this issue, which is more stuck to the core contributions of this work. Specifically, we experimentally find that both implemented methods exhibit the same trend of degradation levels across tasks, RGBT>RGBD>RGBE, which means the performance on RGBT benchmark drops the most after unification. Based on this, to obtain a better unified model, we suggest not placing RGBT task at last. It is also demonstrated by the performance on UniBench300. As shown in Table 4, variants with RGBT task placed at last always produce worse performance. Besides, according to the quantitative results, we offer a recommendation for the specific sequence of task unification, which is RGBT-RGBD-RGBE. Accordingly, we will involve the corresponding discussions in the final version.

References

[1] Bing Cao, Junliang Guo, Pengfei Zhu, and Qinghua Hu. 2024. Bi-directional adapter for multimodal tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 927–935.

[2] Shuang Gong, Zhu Teng, Rui Li, Jack Fan, Baopeng Zhang, and Jianping Fan. 2024. MINet: Modality interaction network for unified multi-modal tracking. *Image and Vision Computing* 148 (2024), 105071.

[3] Lingyi Hong, Shilin Yan, Renrui Zhang, Wanyun Li, Xinyu Zhou, Pinxue Guo, Kaixun Jiang, Yiting Chen, Jinglun Li, Zhaoyu Chen, et al. 2024. Onetracker: Unifying visual object tracking with foundation models and efficient tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19079–19091.

[4] Xiaojun Hou, Jiazheng Xing, Yijie Qian, Yaowei Guo, Shuo Xin, Junhao Chen, Kai Tang, Mengmeng Wang, Zhengkai Jiang, Liang Liu, et al. 2024. Sdstrack: Self-distillation symmetric adapter learning for multi-modal visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26551–26561.

[5] Chenglong Li, Xinyan Liang, Yijuan Lu, Nan Zhao, and Jin Tang. 2019. RGB-T object tracking: Benchmark and baseline. *Pattern Recognition* 96 (2019), 106977.

[6] Chenglong Li, Wanlin Xue, Yaqing Jia, Zhichen Qu, Bin Luo, Jin Tang, and Dengdi Sun. 2021. LasHeR: A large-scale high-diversity benchmark for RGBT tracking. *IEEE Transactions on Image Processing* 31 (2021), 392–404.

[7] Chang Liu, Ziqi Guan, Simiao Lai, Yang Liu, Huchuan Lu, and Dong Wang. 2024. EMTrack: Efficient Multimodal Object Tracking. *IEEE Transactions on Circuits and Systems for Video Technology* (2024).

[8] Lei Liu, Chenglong Li, Yun Xiao, and Jin Tang. 2023. Quality-aware rgbt tracking via supervised reliability learning and weighted residual guidance. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3129–3137.

[9] Meng Sun, Xiaotao Liu, Hongyu Wang, and Jing Liu. 2024. MixRGBX: Universal multi-modal tracking with symmetric mixed attention. *Neurocomputing* 603 (2024), 128274.

[10] Yuedong Tan, Zongwei Wu, Yuqian Fu, Zhuyun Zhou, Guolei Sun, Chao Ma, Danda Pani Paudel, Luc Van Gool, and Radu Timofte. 2024. Towards a generalist and blind RGB-X tracker. *arXiv preprint arXiv:2405.17773* (2024).

[11] Zhangyong Tang, Tianyang Xu, Xiaojun Wu, Xue-Feng Zhu, and Josef Kittler. 2024. Generative-Based Fusion Mechanism for Multi-Modal Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 5189–5197.

[12] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[13] Xiao Wang, Jianing Li, Lin Zhu, Zhipeng Zhang, Zhe Chen, Xin Li, Yaowei Wang, Yonghong Tian, and Feng Wu. 2023. Visevent: Reliable object tracking via collaboration of frame and event flows. *IEEE Transactions on Cybernetics* (2023).

[14] Zongwei Wu, Jilai Zheng, Xiangxuan Ren, Florin-Alexandru Vasluianu, Chao Ma, Danda Pani Paudel, Luc Van Gool, and Radu Timofte. 2024. Single-model and any-modality for video object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19156–19166.

[15] Song Yan, Jinyu Yang, Jani Käpylä, Feng Zheng, Aleš Leonardis, and Joni-Kristian Kämäräinen. 2021. Depthtrack: Unveiling the power of rgbd tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10725–10733.

[16] Jinyu Yang, Zhe Li, Feng Zheng, Ales Leonardis, and Jingkuan Song. 2022. Prompting for multi-modal tracking. In *Proceedings of the 30th ACM international conference on multimedia*. 3492–3500.

[17] Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. 2023. Visual Prompt Multi-Modal Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9516–9526.