SAM2Auto: Auto Annotation Using FLASH

Arash Rocky, *Graduate Student Member, IEEE,* Q. M. Jonathan Wu, *Senior Member, IEEE,*

Fig. 1. 图像 [1] 中详细标注的一个例子。

在过去的几年中， MAGE 处理和计算机视觉社区取得了显著的进展。大型语言模型（LLMs）和视觉语言模型（VLMs）的出现已经触及了人类生活的各个方面，从撰写电子邮件的辅助到解决科学问题；这些模型展示了它们的能力，并且仍在改进的过程中。

尽管如此，在对比大型语言模型（LLMs）和视觉语言模型（VLMs）时，后者的进展未能跟上前者的步伐，主要原因在于它们所训练的数据集的性质。LLMs 可以利用互联网中多种多样且容易获取的来源的大量文本输入，如书籍、文章和代码库，从而实现大规模训练并减少人为干预。这样的丰富性显著增强了它们的性能和可扩展性。相比之下，VLMs 需要采用将视觉输入（图像和视频）与相应文本注释（如图像说明或对象标签）配对的数据集。创建这些数据集劳动强度大、成本高且耗时，导致用于 VLM 训练的高质量、大规模注释数据集的相对匮乏。注释过程通常需要人工努力或先进工具，这对 VLMs 的可扩展性施加了额外限制。

此外，VLMs 的跨学科性质，要求对两种模态（视觉和语言）进行对齐和桥接，这使得它们的训练相比仅在文本单一模态中操作的 LLMs 更加复杂。这种复杂性需要创新的架构，如双编码器或交叉注意力机制，以有效地对齐视觉和文本表示。尽管最近的进展，如开发出像 LAION 这样的网络规模数据集以及自监督学习技术（如 CLIP 和 DINO），已经提高了 VLMs 的可扩展性，但由于用于训练的文本数据的丰富性和多样性无与伦比，VLMs 仍然落后于 LLMs。然而，视觉语言模型的进展，如 CLIP、BLIP 和 Flamingo 等研究成果，表明在缩小这一差距上具有很大潜力。

Manuscript received ****, ***; accepted *****. Date of publication *****; date of current version *****. Paper no. *****. This work was partially supported by the TrustCAV, a CREATE program of the Natural Sciences and Engineering Research Council of Canada.

A. Rocky and Q. M. J. Wu are with the Department of Electrical and Computer Engineering, University of Windsor N9B 3P4, Canada. (Corresponding author: Q. M. J. Wu. e-mail: { rocky, jwu } @uwindsor.ca).

A. 问题定义

视觉 Transformer（ViTs）在计算机视觉任务中显示出显著的发展，然而，[2] 和 [3] 等研究指出了一个关键瓶颈：对大规模标注数据集的需求。这个需求暴露了计算机视觉中一个尚未解决的基本挑战：数据集标注。

创建高质量的注释数据集面临着多重挑战。人工注释是一个耗时的过程，可能会持续数月，并且固有地易受到注释者偏差和不一致的影响。雇佣专家注释者或使用数据标注平台的财务负担可能会很大，特别是对于需要领域专长的特定领域。当处理复杂场景时，例如标注小对象或为连续帧创建精确的真实值注释，这些挑战将进一步放大，如图 1 所示。此外，如 [4] 中所示，注释通常需要针对特定的方法学要求进行调整，迫使研究人员重复重新注释数据集以与其具体研究需求对齐——这一重复且资源密集的过程阻碍了研究进展。

在机器学习和计算机视觉的背景下，标注是指向原始数据（例如图像、视频或文本）添加结构化标签或元数据的过程，以使其可用于训练机器学习模型。这些标注定义了模型学习预测或分类的真实情况。尽管文本标注不在本文的讨论范围内，我们将重点关注图像和视频标注的相关方面。

图像标注涉及标记视觉数据，以训练用于对象识别和理解的模型。常见的类型包括用于勾勒对象的边界框、用于精确形状的多边形分割，以及用于标记每个像素的语义分割 [5] 。此外，实例分割区分同一类别的多个实例，而关键点标注 [6] 标记诸如面部特征点或身体关节等特征。此外，3D 标注在三维空间中标记对象 [7] ，对于自动驾驶等任务至关重要。

视频标注将图像标注扩展到连续帧，关注动态元素。关键类型包括目标跟踪，它在各帧之间保持一致的标识（ID），以及动作识别，识别“行走”或“跳跃”等活动。事件检测则标注高级别的事件（例如，“交通事故”[4] ），而运动跟踪则分析物体轨迹。这些标注对于监控和自动化系统 [7] 等应用至关重要。

本研究为视频注释提供了一个基线，重点关注在具有一致 ID 的序列中对象的边界框。我们认为，自动化这种类型的注释可以为轻松生成其他注释类型铺平道路。

为了解决这个缺陷，已经提出了许多策略。[8] 和 [9] 的研究人员探索了自监督学习方法，旨在减少对标签数据的依赖。此外，[10] 引入的在线标记生成方法，以及 [11] 研究的数据增强结合正则化技术的方法，也被用于提高训练过程中的数据效率。

在应用层面上，创新的方法已经出现，以进一步减少对人工标注的依赖。例如，[12] 建议使用由潜在扩散模型生成的合成数据进行无标注的物体计数，并结合在有序图像三元组上训练的排序网络。类似地，[13] 使用预训练的二维模型和对比学习来检测和分类点云中的三维物体，而无需人工的三维标注。此外，Emergent Spatial-Temporal Scene Decomposition via Self-Supervision (EmerNeRF) [14] 介绍了一种将场景分解为静态和动态组件的自监督方法，该方法消除了对真实标注或预训练模型的需求。作为避免三维

标注的另一尝试，Stereo4D [15] 从互联网上的立体视频中提取高质量的动态三维运动和长期运动轨迹，以创建一个大规模、真实世界的四维场景数据集。

B. 解决方案：自动标注

正如在 **??** 中讨论的那样，视觉-语言模型（VLMs）的发展受到标注数据集稀缺的限制。虽然研究人员提出了创造性的方法来解决这一限制 **??** ，该领域仍然缺乏一个强大的自动标注方法，可以以最少的人为干预产生高质量的数据集。

我们提出解决方案在于利用开放词汇物体检测方法 [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31] 。这些检测器可以为已知和未知的物体类别生成高置信度的边界框。然而，关键挑战在于在视频帧之间保持一致的物体 ID——这项任务需要可靠的物体跟踪，最少的 ID 转换，并且在检测轨迹和真实轨迹之间实现高交并比（IoU）。需要注意的是，如果实现了帧序列（即整个数据集）的自动标注，单帧图像数据集的问题也将得到有效解决。近期关于 SAM 2：在图像和视频中分割任何东西 [32] 的研究在这个方向上取得了进展，通过引入可跟踪性特征。他们的方法允许人工标注者提供第一帧的标注，并通过正负点击来修饰后续帧。虽然这减少了人工操作，但仍然需要人工干预。我们对开放世界物体检测模型的广泛实验揭示了一个关键限制：虽然这些模型理论上可以在无需额外训练的情况下检测目标物体，但它们的性能严重依赖于帧特定的超参数调整。在整个数据集或视频序列中使用一组参数，会导致大量误报或漏报，尤其是在动态背景场景中。为解决这些挑战，我们引入 SAM2Auto——一个结合两个关键组件的自动标注流程：

1) SMART-OD：一个稳健的目标检测系统，它结合了 SAM2 [32] 用于自动掩膜生成、YOLO-World [17] 用于开放世界目标检测，并通过 SAHI [33] 提高准确性。我们的系统采用统计方法来在跟踪前最小化误报。
2) FLASH（F rame- L evel A nnotation and S egmentation H andler）：一个多物体追踪器（MOT），可以扩展任何基于记忆的分割模型的可追踪特性，以在各帧之间保持一致的对象识别，从首次检测到最后出现，无论检测间歇性中断。

该方法提供了几个主要优势：它消除了手动标注的需要，确保了跨帧的一致对象识别，并且无需特定于数据集的训练或广泛的参数调整即可运行。我们的实验结果表明，SAM2Auto 可以在显著较少的时间内自动生成高质量的标注，同时保持与手动方法相当的准确性。本文的主要贡献如下：

1) 我们引入了 SAM2Auto，这是第一个完全自动化的视频数据集标注流水线，不需要人工干预或特定数据集的训练
2) 我们介绍了 SMART-OD，一种结合了 SAM2、YOLO-World 和 SAHI 的新型目标检测系统，可以在不同场景中实现一致且准确的目标检测
3) 我们开发了 FLASH，这是一种多对象实时视频实例分割（VIS）方法，即使在检测间隙中断的情况下也能维护对象在帧之间的身份
4) 我们通过大量实验表明，SAM2Auto 的准确性与人工标注相当，同时显著减少了标注时间和成本
5) 我们提供了全面的消融研究，展示了每个组件的有效性及其在流程中的集成

在提供了自动标注的总体概况后，本文的其余部分组织如下：在第 I 节，讨论了与自动标注主题相关的相关工作。第 II 节包含对象检测方法 SMART-OD、对象跟踪器 FLASH 和整体 SAM2Auto 架构。第 III 节提供了实验结果和相关讨论。未来方向在第 IV 节中进行讨论。最后，本文在第 **??** 节中作出总结。

I. 相关工作

A. 自动注释的正确物体检测

多种开放词汇物体检测方法推进了自动标注技术，每种方法在特定领域都有其优势。YOLO-World [17] 扩展了 YOLO 系列，通过视觉-语言建模实现对未见物体的高效零样本检测。其实时性能（在 LVIS 上以 52 FPS 达到 35.4 AP）使其成为大规模标注的理想选择。DINO-X [20] 提供了强大的开放世界检测并具有优越的泛化能力，但需要更高的计算成本，限制了其实时可用性。Sapiens [18] 专注于以人为中心的任务，如姿态估计，在高分辨率人类标注方面表现优异，但缺乏普适性。RT-DETR [19] 以多尺度处理在精度和速度间取得平衡，但在实时效率上不如 YOLO-World。总的来说，我们将 YOLO-World 集成到我们的物体检测模型中，因为凭借其速度、适应性和零样本能力，它是可扩展、实时自动标注的最实用选择。

B. 自动注释中光照变化的处理

光照条件的变化和多样的天气条件对物体检测系统提出了重大挑战。传统方法通过使用像 BDD100K [34] 和 nuScenes [35] 这样的数据集进行数据增强，或通过像 MFNet [36] 这样的多模态传感器融合来解决这个问题。低光增强技术已经成为关键的预处理方法，解决方案包括基于神经网络的方法，如 MIRNet [37] 和 RUAS [38] ，以及像 Zero-DCE [39] 这样的零参考方法。域适应技术，例如 CycleGAN [40] ，也已经被探索用于适应在日间图像上训练的模型以适用于夜间条件。虽然这些技术改善了检测性能，但它们通常需要大量的训练数据或计算密集的预处理。我们的方法利用 SAM2 [32] 进行物体检测前的自动图像分割，无需额外训练，同时有效处理光照和天气变化。在不需要特定领域适应的情况下，简化了流程，同时在各种条件下保持了稳健的性能。

C. 自动标注中的稳健跟踪

跟踪在计算机视觉和图像处理领域中取得了显著的进展。虽然传统的多目标跟踪 (MOT) 仍然是一个具有挑战性的研究领域 [41], [42], [43], [44], [45], [46], [47], [48], [49] ，许多研究已经将其范围扩展到多目标跟踪和分割 (MOTS) [50] 、视频对象分割 (VOS) [51], [52], [53] 、交互式视频对象分割 (iVOS) [54], [55], [56] 和视频实例分割 (VIS) [57], [58] 。随着点跟踪器的出现，跟踪变得更加多样化，这是由三维对象重建中对精确关键点对应关系的需求驱动的。如果我们将这种多样性概括为视频中的任意跟踪，我们可以将跟踪方法分类如下：MOT 跟踪器基于物体的边界框，MOTS、VOS、iVOS 和 VIS 的跟踪器依赖于物体的像素级掩码，而点跟踪器则专注于物体的细粒度关键点。

尽管在每个类别中缺乏大规模的标注数据集 [41], [47], [32], [59] ，跟踪方法在很大程度上仍独立于其他类别的数据集，且它们的方法之间几乎没有收敛。为了解决这个问题，OmniTracker [60] 提出了一种统一的跟踪与检测架构，

以整合边界框跟踪器（MOT）和掩码跟踪器（VOS, iVOS, VIS）。然而，这种方法需要针对特定数据集进行训练，使其在自动注释时不切实际。

为了应对这一限制，SPAM [61] 被开发用于自动标注，利用合成预训练、伪标签和基于图模型的主动学习。虽然它主要侧重于 MOT 标注，但仍需有限的人为监督和额外的训练。对于 VOS 和 iVOS，SAM2 [32] 引入了一种基于记忆库的方法来提高遮罩跟踪的效率。虽然 SAM2 可以跟踪边界清晰的对象，包括子部分，但由于内存不足的问题，它在处理长视频序列和大量对象时表现不佳。值得注意的是，近期关于 SAM2 [62], [63], [64], [65], [66] 的研究中没有一个解决了这一限制问题。

为了弥合这些差距，我们引入了 SAM2ASH，这是一种用于标注的统一边界框（MOT）和基于掩码的（VOS, iVOS, VIS）追踪器，它扩展了 SAM2 的可追踪性，同时消除了内存限制。与现有的基于内存库的追踪器不同，SAM2ASH 可以高效处理长视频序列和多个对象。此外，它不需要额外的训练，利用丰富的 SAM2 检查点实现稳健的性能。

D. 自动标注综述

是自动注释领域最早的研究之一，[67] 使用了多实例学习（MIL）作为图像分类和文本注释对齐的弱监督方法。这种方法是最早认识到由于大规模未标记数据集的缘故，全监督是不现实的研究之一。同样，成本效益标注 [68] 研究了注释成本和模型性能之间的权衡，提供了有关高效数据集管理的见解。虽然基于 MIL 的方法依赖于间接监督，但成本感知方法则着重于在有限的注释资源下平衡性能。这些方法与完全自动化的视觉语言模型形成对比，后者旨在通过利用大规模预训练来完全消除手动标注。

最近在视觉-语言模型方面的进展使得可以进行自动注释而无需预定义类别。[69] 和 [70] 使用视觉-语言预训练来检测传统标记数据集之外的物体。与依赖预定义对象类别的弱监督方法不同，这些模型可以超越有限的注释进行泛化。扩展这一点，自动像素级开放词汇实例分割 (APOVIS) [71] 和实时 VIS [72] 将开放词汇能力扩展至静态和视频场景中的实例分割。虽然视觉-语言模型旨在实现零样本或少量样本注释，减少人工参与，但它们依赖于大规模预训练可能引入潜在的偏差和不准确性，尤其是在应用于特定领域任务如自动驾驶时。

虽然视觉语言模型实现了广泛的泛化，但自动驾驶应用需要特定领域的标注技术。自动数据引擎（AIDE）用于自动驾驶中的目标检测 [73] 引入了一个自训练框架，该框架通过迭代优化标注，随着时间的推移减少错误。同样，[74] 专注于基于 LIDAR 的目标标注，根据运动轨迹完善标注。与依赖大规模互联网数据的开放词汇模型不同，这些方法集成了传感器数据以提高特定领域的准确性。然而，这种任务特定的方法限制了其在不同数据集上的泛化能力，这是视觉语言模型更有效解决的挑战。

除了学术研究驱动的方法外，业界的工具如 “Cosmos World Foundation Model Platform for Physical AI [75] ” 和 “Auto Labeling Images with Roboflow [76] ” 都集成了基础模型以在商业应用中实现自动标注。这些平台提供可扩展的标注管道，但往往依赖于专有数据和基础设施。相比之下，SAM2 的数据引擎 [32] 通过三个关键阶段演变以改进视频分割标注：(1) 使用原始 SAM 的逐帧标注，(2) 通过 SAM2 Mask 进行时间掩码传播，以及 (3) 完全整合 SAM2，该系统结合了时间记忆和多提示支持。该系统支持连续的模型再训练、质量验证和自动 masklet 生成，确保多样且高质量的标注。通过利用人机互动的方法，SAM2 创建了一个不断提高效率和标注质量的良性循环。与依赖于封闭数据集的传统工业工具不同，SAM2 在时间一致性和自动化方面的开放整合使其特别适合大规模和高精度的视频分割任务。

继工业解决方案之后，最近的学术研究探索了半监督方法作为使标注更自动化的替代方案。半监督开放世界物体检测 [77] 和 SPAMming Labels [61] 利用半监督以最少的人为干预来优化标签。与依赖有限的监督信号的弱监督和领域特定标注方法不同，半监督方法使用标记和未标记的数据动态优化标注。虽然它们在标注成本和模型准确性之间取得平衡，但它们仍然不如开放词汇模型灵活，后者可以无需重新训练即可检测到未见过的对象。

尽管取得了这些进展，现有的方法在准确性、效率或适应性方面仍然存在困难。为了解决这些局限，我们引入了 SAM2Auto，这是一种新颖的框架，结合了快速且精确的开放词汇物体检测系统与像素级开放词汇视觉实例分割（VIS）框架。SMART-OD 作为一种强大的开放词汇物体检测器，能够适应光照变化，而 FLASH 作为一种实时多目标跟踪（MOT）的 VIS，能够跨帧恢复不一致的检测以生成高度准确的实例级标注。SAM2Auto 的名称反映了 SAM2 在检测和跟踪中的双重用途，强化了其在自动化标注中的作用。

II. 自动标注方法

在本节中，我们介绍了一种新颖的自动标注流程，称为 SAM2Auto。它集成了 SMART-OD（一种开放词汇的目标检测模型）和 FLASH（一种多目标实时视频实例分割（VIS）模型），旨在实现经济高效且高度准确的标注。通过结合这些组件，SAM2Auto 在保证大规模数据集的效率的同时，实现了高质量的实例级标注。以下小节详细介绍了每个组件及其在整个系统中的作用。

A. SMART-OD: 稳健的开放词汇检测的数据集优化流程

我们提出了一种多阶段的目标检测和验证流程，该流程将开放词汇检测能力从单个图像推广到整个序列和数据集。我们称这种方法为用于目标检测的分割、掩码引导分析和稳健阈值（SMART-OD），它解决了当前开放词汇方法中的一个关键限制：这些方法通常在单个图像上表现出色，但在视频序列中的不同帧或数据集内的多样化图像中难以保持一致性和准确性。如图 2 所示，SMART-OD 流程由三个连续的阶段组成，逐步细化检测过程，如下所述：

1) 通过自动掩模生成进行分割： SMART-OD 的第一阶段使用一个基础模型进行视觉分割，该模型擅长识别物体边界，我们应用了 SAM2 [32] 。分割模型为场景中所有潜在物体生成实例掩码，无需语义分类，这将具有不同光照条件的输入图像转化为统一的掩码表示，从而在多样的成像条件中建立一致的视觉表现。

2) 目标检测中的掩膜引导分析： 在 SMART-OD 的第二阶段中，我们实现了一种开放词汇物体检测（OVOD）方法，该方法能够在每个数据集的所有序列和帧中一致地识别物体。如 I-A 所讨论的，我们选择了 YOLO-World [17] 作为最适合的 OVOD，基于它在面向掩码输入时的速度、适应性和零样本能力。

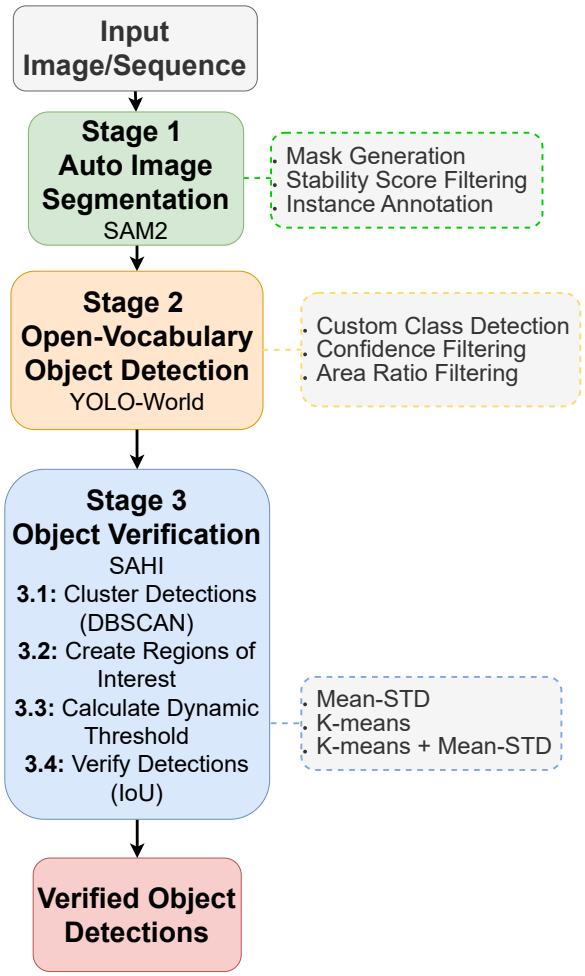

Fig. 2. SMART-OD: 数据集优化管道用于鲁棒的开放词汇检测。系统通过三个主要阶段处理输入图像：使用 SAM2 进行分割，使用 YOLO-World 进行面罩引导分析，以及使用基于 SAHI 的验证和动态阈值进行鲁棒阈值化。

3) 检测验证中的稳健阈值化： 虽然 YOLO-World 提供了高效的开放词汇检测，但对其参数进行优化以检测各种数据集中大多数对象仍然具有挑战性。在 SMART-OD 管道中，我们通过实施一个稳健的验证阶段来解决这个问题，该阶段在保持高召回率的同时抑制不可避免的错误检测。这个阶段通过局部处理采用 SAHI [33] 进行验证，并包含几个关键组件：

3.1：检测聚类。我们使用 DBSCAN [78] 聚类空间上接近的检测以创建感兴趣区域（ROIs）：

$$C = \text{DBSCAN}(B, \epsilon, \mu) \tag{1}$$

，其中 C 表示簇，B 是边界框集合，ϵ 是点之间的最大距离，μ 是簇中的最小样本数。对于每个簇，我们创建一个合并的 ROI：

$$R_i = [\min_{b\in C_i} x_1, \min_{b\in C_i} y_1, \max_{b\in C_i} x_2, \max_{b\in C_i} y_2] \tag{2}$$

，其中 R_i 是簇 C_i 的 ROI，(x_1, y_1, x_2, y_2) 是簇中每个边界框 b 的坐标。

3.2: 动态阈值。作为阶段 3 的关键步骤，SMART-OD 流程通过自适应阈值抑制误报，该阈值对每个帧的置信度分布做出响应。动态阈值 θ_d 通过四种方法之一进行计算，然后由最小阈值 $\theta_{\min}$ 进行限制：

$$\theta_d = \begin{cases} \mu_s - \sigma_s & \text{(Mean-STD)} \\ \min_{s\in S_t} s & \text{(K-means)} \\ \mu_l + 2\sigma_l & \text{(K-means + Mean-STD)} \\ \text{Two-stage clustering} & \text{(Double K-means)} \end{cases} \tag{3}$$

$$[0.5em]\theta_{\text{final}} = \max(\theta_d, \theta_{\min}) \tag{4}$$

其中，μ_s, σ_s 是所有置信度得分的均值和标准差，S_t 代表前两个 K 均值聚类中的得分，而 μ_l, σ_l 是来自最低置信度聚类的统计信息。

3.3: ROI 处理与验证

使用 SAHI 方法对每个 ROI 进行单独处理。要验证一个检测，它必须满足两个标准：

1) 检测 IoU：检测必须与 SAHI 预测具有高于验证阈值的 IoU：

$$\max_{p\in P} \text{IoU}(b, p) > \theta_v \tag{5}$$

其中 P 是 SAHI 预测集，b 是检测框，而 θ_v 是验证 IoU 阈值。

2) 检测置信度：检测置信度必须超过基于公式 3 的动态阈值：

$$c_b > \theta_{\text{final}} \tag{6}$$

其中 c_b 是检测 b 的置信度。

这种双重标准方法确保只有具有强验证支持的高置信度检测才会被保留为 SMART-OD 的最终检测结果。处理数据集每个序列的 SMART-OD 详细公式和综合算法在附录 A 中提供。

B. FLASH：用于自动标注的实时视觉系统

我们介绍了 FLASH（帧级注释和分割处理器），这是一个通过利用基于记忆的分割模块，将多目标跟踪与高质量分割相结合的框架。FLASH 将边界框检测转换为详细的多边形分割，同时保持跨帧的时间一致性。

FLASH 由三个主要模块组成，如图 3 所示：

1) 初始化模块：处理检查点加载和系统准备
2) 在线对象关联模块：将检测与在线对象追踪关联起来
3) 注释与分割处理器 (ASH)：管理基于内存的分割模型

1) 初始化模块： 初始化模块作为 FLASH 框架的入口，负责系统准备，并为处理长视频序列提供强大的恢复能力。此模块执行三个关键功能：

- 检查点管理：加载先前保存的处理状态并处理当前状态的序列化
- 恢复功能：允许在中断后从特定帧继续处理
- 内存高效处理：实现自适应块选择机制，以处理超出内存限制的长视频，利用时间帧优化来保持一致性

检查点管理协议和恢复能力算法的详细实现见附录 B。为了解决在处理 ASH 模块中的长视频时的内存限制问题，初始化模块实施了自适应块选择机制。该系统通过最佳帧识别来保持时间一致性，选择帧的标准如下：

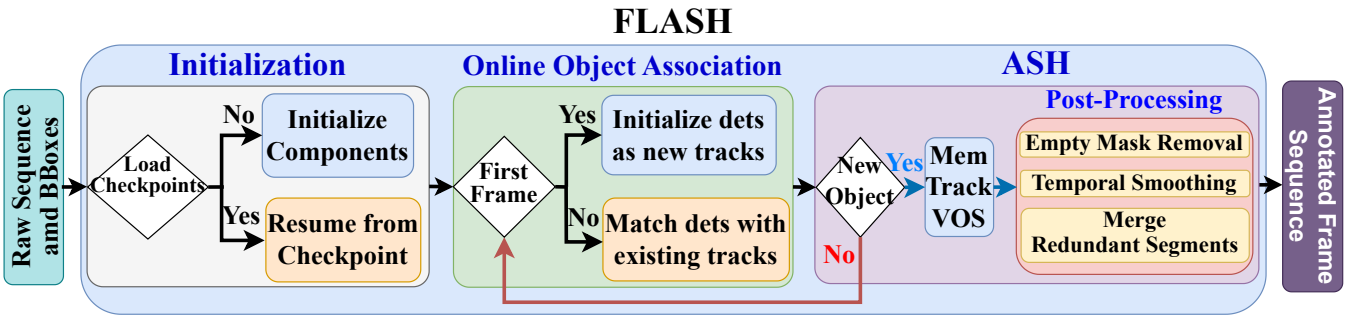

Fig. 3. FLASH 系统架构展示了三个主要模块：初始化、在线对象关联以及注释与分割处理器（ASH）。它接收原始帧序列和边框（BBoxes），并输出带有基于记忆的分割（Mem Track VOS）核心的注释帧序列。

$$\text{OptimalFrame} = \arg\max_{f} |\mathcal{O}_f| \text{ such that } c - w \le f \le c + w \quad (7)$$

其中，$\mathcal{O}_f$ 表示帧 f 中的对象集合，c 是当前帧，w 是窗口大小。

2) 在线目标关联模块： 在线目标关联模块构成了 FLASH 框架中目标检测与分割之间的关键桥梁。该模块通过系统地将检测框与一个在线目标追踪器关联，作为我们的多目标追踪（MOT）解决方案，并辅以基于 IoU 的关联，以保持目标身份的一致性，从而维护时间一致性。
模块执行三个基本功能：

- 时间上下文处理：区分第一帧和后续帧处理方式
- 检测-轨迹关联：将新检测到的物体映射到现有轨迹
- 新对象识别：确定哪些检测需要分割初始化

时间上下文处理和检测轨迹关联算法的详细实现见附录 B 。

新物体识别：在处理完所有关联后，任何未匹配到现有轨迹的检测都被识别为需要轨迹初始化的新物体。形式上，帧 F_j 的新物体集 N_j 定义为：

$$N_j = \{d_i \in D_j : i \notin \text{dom}(\mathcal{M})\} \quad (8)$$

，其中 $\text{dom}(\mathcal{M})$ 表示映射函数 $\mathcal{M}$ 的定义域。

对于每一个未匹配的检测，系统分配一个新的唯一轨迹 ID，初始化一个新的轨迹，并将此信息加入已知轨迹集。这种系统化的方法确保所有对象在整个视频序列中都被正确地跟踪，同时保持各自的独特身份。这些新的对象将传递给 ASH 模块以为当前帧的整个序列创建其注释和分割。

3) 注释和分割处理器 (ASH)： FLASH 的核心是注释和分割处理器（ASH），它通过在线对象关联模块初始化新对象的边界框，并使用基于记忆的分割模型来传播整个序列中所有这些新对象的掩码分割。这创造了一个多对象实时视频实例分割（VIS）模型。该架构有效地处理了保持时间一致性、管理计算资源以及在准确性和实时性能要求之间取得平衡的复杂挑战。

ASH 模块逐帧处理视频序列，同时在时间边界上保持对象身份，生成基于多边形的分割掩膜和边框表示。
该模块执行三个基本功能：

- 内存高效处理：优化帧和对象处理以提高计算效率（详细信息可参见附录 B ）
- 将 VOS 转化为实时 VIS：将单目标分割转换为多目标跟踪
- 后处理：增强分割质量和一致性

将 VOS 转换为实时 VIS：ASH 通过两个关键机制将单对象视频分割能力扩展到多对象跟踪。

与在线对象关联的集成：分割模型需要为新对象的边框提供唯一的对象 ID。为此，ASH 利用了从在线对象关联模块分配给相关对象的跟踪 ID。更具体地，新对象集合 $\mathcal{N}_t = \{o_1^t, o_2^t, ..., o_n^t\}$ 也代表了在线对象关联模块中被跟踪对象的集合，其中每个对象 o_i^t 都与一个唯一的跟踪 ID 相关联。结果，对于在帧 t 中检测到的新对象：

$$o_i^t = \begin{cases} \text{track}_j & \text{if } \exists j : \text{IoU}(b_i^t, b_j^{t-1}) > \tau_{\text{track}} \\ \text{ID}_{\text{new}} & \text{otherwise} \end{cases} \quad (9)$$

其中，b_i^t 是在帧 t 上对象 i 的边界框，τ_{track} 是跟踪阈值，$\text{ID}_{\text{new}} = \max(\text{IDs}) + 1$ 用于新对象的分配。这样的方法在各帧之间保持一致的对象身份，这是跟踪精度的关键组成部分。

多目标掩膜传播和表示转换：新物体被输入分割模型，该模型在整个序列中生成物体的二元掩膜。对于每个在第 t 帧初始化的物体 o_i^t ，分割模型在后续帧中生成掩膜：

$$M_i^{t+\delta} = \phi(o_i^t, F_t, F_{t+\delta}), \quad \delta \in \{0, 1, ..., (T-t)\} \quad (10)$$

其中 ϕ 表示基于记忆的分割函数，它将初始对象传播到后续帧。

我们的模块通过将二进制掩码转换为多边形表示，实现了分割结果的双重表示策略：

$$P_i^j = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\} = \psi(M_i^j) \quad (11)$$

其中 ψ 是识别掩码边界点的轮廓提取函数。这种转换提供了显著的存储效率，并有助于可视化和分析等后续处理任务。

后处理：ASH 应用三种关键的后处理技术来提高分割质量：

空掩膜移除：我们首先识别每个对象 $\tau(o_i)$ 的时间终点，作为对象保持有效掩膜表示的最后一帧：$\tau(o_i) = \max\{t \in [1, T] : \sum_{x,y} M_i^t(x, y) > \epsilon\}$ ，其中 ϵ 是有效掩膜内容的最低阈值。随后，我们通过应用时间一致性过滤器 ϕ 在整个序列中进行掩膜修剪，该过滤器定义为：

$$\phi(M_i^t) = \begin{cases} M_i^t & \text{if } t \le \tau(o_i) \\ \emptyset & \text{if } t > \tau(o_i) \end{cases} \quad (12)$$

这有效地消除了在物体退出场景后出现的虚假遮罩。然后计算出细化的分割集 $\hat{\mathcal{M}}$ ，如下所示：$\hat{\mathcal{M}} = \{\phi(M_i^t) : \forall i \in [1, n], t \in [1, T] \text{ where } \phi(M_i^t) \neq \emptyset\}$

时间平滑：当分割模型分别处理每一帧时，它们可能由于光照、视角或物体变形的细微变化而产生抖动、闪烁的边界，尽管实际物体移动是平稳的。这种视觉不稳定不仅降低了分割的感知质量，还可能对对象跟踪等下游应用产生负面影响。通过应用时间平滑，在数学上表示为当前帧和先前帧分割的加权平均（13），系统可以抑制高频噪声，同时保留合法的运动。这种平滑通过减少由边界变化异常引起的错误关联改善了跟踪准确性，并增强了系统在单个帧中临时分割失败情况下的整体稳健性。

$$\hat{P}_i^t = \alpha \cdot P_i^t + (1-\alpha) \cdot P_i^{t-1} \tag{13}$$

其中 $\alpha \in [0,1]$ 是时间平滑因子。

冗余片段的合并：当分割掩码在视频帧之间传播时，由于对象变形、遮挡、光照变化、模型限制以及在线对象关联模块在不同数据集和序列中的广义配置等因素，它们往往会累积错误。这些挑战可能导致多个分割实例代表同一个物理对象——特别是在对象分裂、合并或临时跟踪失败期间。此类冗余增加了计算和存储成本，并可能在最终输出中引入视觉伪影。为了解决这一问题，ASH 执行冗余片段的合并，基于交并比 (IoU) 阈值 τ_{merge} 合并具有高度重叠的片段。这显著提高了对象表示的视觉质量和准确性。该方法在概念上类似于目标检测中的非极大值抑制，但扩展到了时空域，确保每个对象在整个视频中都由单一、高质量的分割掩码表示。因此，经过改进的一组多边形计算为：

$$\hat{\mathscr{P}} = \left\{ \hat{P}_i^t : \forall i,t \text{ such that } \max_{j \neq i} \left(\text{IoU}(\hat{P}_i^t, \hat{P}_j^t) \right) < \tau_{\text{merge}} \right\} \tag{14}$$

，其中 τ_{merge} 定义了合并的重叠阈值。结合时间一致性执行和噪声降低，这个后处理步骤显著增强了分割质量，并有益于如对象跟踪等下游任务。

C. 用于长序列的鲁棒块处理

FLASH 实现了一种自适应分块策略，以处理超过内存限制或需要稳健回退机制的长视频序列。这种方法将视频处理划分为可管理的片段，同时在块边界维持时间一致性。

基于块的处理子系统由两个主要组件组成：

- 块管理器：协调将长序列划分为重叠块
- 块处理器：处理具有边界一致性的单独块处理

块管理器：系统首先尝试使用统一处理处理完整序列。如果由于内存限制或计算限制而失败，它会自动切换到基于块的处理。块管理器采用滑动窗口方法，具有可配置的块大小和重叠参数：

$$\mathscr{C} = \{C_1, C_2, ..., C_n\} \text{ where } C_i = [s_i, e_i] \tag{15}$$

其中 $\mathscr{C}$ 代表块的集合，C_i 代表第 i 个块，其跨越帧 s_i 到 e_i ，并且块保持时间重叠以确保一致性：

$$s_{i+1} = e_i - \omega \text{ for } i \in \{1, 2, ..., n-1\} \tag{16}$$

其中 ω 表示重叠大小。为了获得最佳的块边界，系统使用以下方法识别高对象密度的区域：

$$\text{OptimalStart}_i = \arg \max_{f \in [s_i - \delta, s_i + \delta]} |\mathscr{O}_f| \tag{17}$$

，其中 $\mathscr{O}_f$ 是框架 f 中的对象集合，δ 定义了搜索窗口大小。

块处理器：对于每个块 C_i ，处理器在处理重叠区域时，特别注意检测-跟踪关联和分割。注释和分割处理器（ASH）在重叠区域中使用前一个块的对象状态进行初始化，以确保连续性：

$$\mathscr{S}_{i+1}(s_{i+1}) = \mathscr{S}_i(e_i - \omega) \tag{18}$$

，其中 $\mathscr{S}_i(f)$ 表示块 i 中帧 f 的分割状态。

块间一致性：对于块之间的重叠区域，FLASH 应用了一种专门的基于多边形的对象合并策略，该策略通过直接匹配分割掩码本身的 IoU 来解决身份冲突，而不是依赖于在线对象关联模块较为宽松的约束。这种在 `merge_overlapping_segments` 函数中实现的方法，计算重叠区域中跨帧的对象多边形之间的精确 IoU：

$$\text{IoU}(O_i^A, O_j^B) = \frac{1}{|F_{\text{overlap}}|} \sum_{f \in F_{\text{overlap}}} \frac{|M(O_i^A, f) \cap M(O_j^B, f)|}{|M(O_i^A, f) \cup M(O_j^B, f)|} \tag{19}$$

，其中 O_i^A 表示来自块 A 的对象 i ，O_j^B 表示来自块 B 的对象 j ，F_{overlap} 是重叠帧的集合，$M(O,f)$ 是帧 f 中对象 O 的掩码。

该系统根据重叠区域的最大平均 IoU 维持块特定对象 ID 与一致的全局 ID 之间的映射：

这种健壮的分块策略使得 FLASH 能够以有限的内存需求处理任意长度的视频序列，同时在整个序列中（包括块边界）保持时间一致性。重要的是，它消除了对特定数据集进行调整的需求

$$\text{GlobalID}(O_i^B) = \begin{cases} \text{ID}(O_j^A) & \text{if } \exists j : \text{IoU}(O_i^B, O_j^A) > \tau_{\text{overlap}} \\ \text{ID}(O_i^B) & \text{otherwise} \end{cases} \tag{20}$$

跟踪参数，使该方法在不同的视频数据集上用于自动标注时更加稳健和具有可推广性。此外，FLASH 引入了一种回退机制，可根据可用的计算资源和视频序列的复杂性自动在全序列处理和基于块的处理之间切换。图 4 展示了在序列管理器架构中集成的 FLASH 整体。

利用 SMART-OD 物体检测框架与 FLASH 视频实例分割系统，我们引入了一种新颖的自动标注流程，称为 SAM2Auto。该流程解决了在大规模数据集上创建高质量实例级标注而尽量减少人工干预的挑战。通过对参数优化和验证的系统化方法，SAM2Auto 在多样化的视觉条件下实现了稳健的性能，同时保持计算效率。我们详细介绍了实现方法，并提供了在任意数据集上部署该系统的结构化工作流。我们的方法结合了用于 SMART-OD 自动图像分割阶段的增强版 Segment Anything Model，即 SAM2 [32] ，以及用于开放词汇检测阶段的 YOLO-World [17] 和用于验证阶段的 SAHI [33] 。对于 FLASH 的在线物体关联模块，我们采用 ByteTrack [43] ，而对于 FLASH 的基于内存的视频实例分割框架，我们采用 SAM2 [32] 。这种集成实现了规模化的成本高效且高度准确的标注。

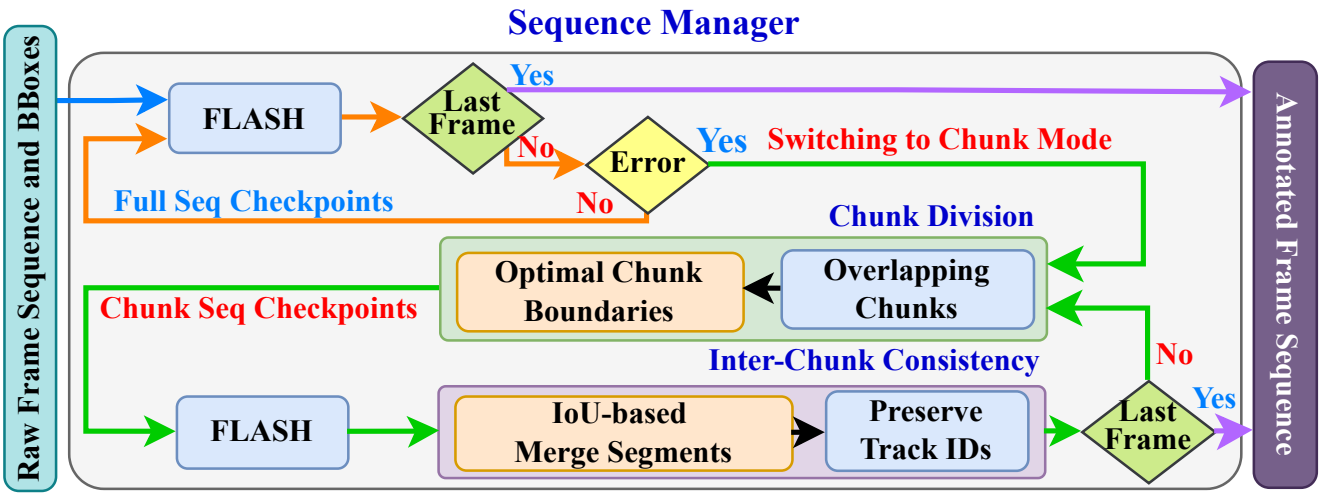

Fig. 4. FLASH 在序列管理器中的集成：通过利用块划分、块间一致性和块序列检查点，FLASH 可以高效地处理任意长度和对象数量的序列而不受限制。

Algorithm 1 SAM2Auto: 系统化自动注释流程

Require: Dataset $\mathscr{D}$, Parameters Θ
Ensure: Annotated Dataset with instance segmentation

$S_{rep} \leftarrow \arg\max_{S \in \mathscr{D}} \max_{f \in S} |O_f|$ ▷ Select sequence with highest object density
$f_{crowd} \leftarrow \arg\max_{f \in S_{rep}} |O_f|$ ▷ Find most crowded frame
$\Theta_{opt} \leftarrow \arg\max_{\Theta} J(\Theta, f_{crowd})$ ▷ Optimize parameters
Apply SMART-OD with Θ_{opt} to S_{rep} and evaluate Precision and Recall
Select random sequence S_{val} for cross-validation
Verify $\min(P_{val}, R_{val}) \geq \gamma \cdot \min(P_{rep}, R_{rep})$
for each sequence S_i in dataset $\mathscr{D}$ **do**
  $D_i \leftarrow \text{SMART-OD}(S_i, \Theta_{opt})$ ▷ Apply object detection
  $V_i \leftarrow \text{SequenceManager}(S_i, D_i, \Theta_{SM})$ ▷ Apply Sequence Manager with FLASH
end for
Perform Quality Assurance on stratified sample of sequences
$Q_i \leftarrow \text{IoU}(V_i, M_i)$ for sampled S_i
for each S_i where $Q_i < \tau_{QA}$ **do**
  Refine parameters and reprocess S_i
end for
return Annotated Dataset with instance segmentation

集成方法：SAM2Auto 采用系统化的七步部署方法，在算法 1 中形式化：(1) 使用最大对象密度进行代表性序列选择，(2) 在最拥挤的帧上使用 SMART-OD 进行参数优化，(3) 序列级验证，(4) 跨序列验证，(5) 数据集范围的检测，(6) 使用 FLASH 进行大规模序列管理应用，(7) 通过分层抽样进行质量保证。

该算法通过识别包含物体数量最多的帧的序列来实现步骤 (1)，然后在这个最具挑战性的帧上进行步骤 (2) 的参数优化。步骤 (3) 将优化后的配置应用于整个代表序列以进行验证，而步骤 (4) 则执行跨序列验证，以确保在数据集的不同部分之间能够广泛适用。步骤 (5) 和 (6) 在主处理循环中处理，其中每个序列在整个数据集范围内进行检测，随后应用序列管理器以及集成的 FLASH 分割。最后，步骤 (7) 通过基于 IoU 的评估和针对低于质量阈值的序列进行有针对性的改进来实施质量保证。详细的七步实施程序在附录 C 中提供。

III. 实验

A. 数据集和指标

数据集。为了确保我们的方法可与近期的跟踪和注释工作相比较，我们将其应用于以下四个著名的目标跟踪数据集：

- MOT17 [79] 是一个标准的跟踪基准，包含 14 个不同的序列，具有不同的摄像机运动、视角和人群密度。根据 [45], [80], [81], [82] ，对于 MOT17 的公共检测，我们利用了 CenterTrack 优化后的边界框。
- MOT20 [83] 是一个具有挑战性的基准，其中包含 8 个高密度视频序列，这些序列是在现实世界中的拥挤场景中捕获的。它强调在极端遮挡、行人间隔紧密和有限的目标可见性下的强大追踪能力，使其成为评估在严重拥堵环境中性能的理想选择。
- DanceTrack [84] 是一个多目标跟踪基准，包含 100 多个标注的舞蹈视频，其中的表演者展示了复杂的动作，伴有频繁的遮挡和同步。与标准的行人数据集不同，它测试了跟踪器在动态运动中保持身份的能力。
- BDD100K [34] 是一个大规模的驾驶视频数据集，包含 100,000 个标注剪辑，捕获于多样的地点、天气和光照条件下。它支持多种任务，包括目标检测、分割、车道检测和多目标跟踪。凭借丰富的标注和广泛的场景覆盖，BDD100K 被广泛用于自动驾驶和多任务学习模型的训练和评估。由于数据集公开性限制，我们使用验证集进行所有方法的评估以确保公平比较。

评估指标。为了从互补的角度评估跟踪质量，确保精准检测和一致的身份保持——这是可靠的自动标注系统的关键，我们通过以下指标评估我们的方法：

- MOTA [85] ： 通过相对于真实目标的错误（误报、漏检、ID 切换数量）来衡量整体跟踪性能。虽然全面，但它强调检测质量却低估了身份保留的重要性。
- IDF1 [86] ： 通过衡量在整个跟踪序列中的一致性来关注身份保留。与 MOTA 不同，它优先考虑一致的身份跟踪，

但可能低估检测的准确性，因为维持较少但一致的跟踪可以获得更高的分数。

- HOTA [87]：一种平衡的度量标准，能够等同地衡量检测准确性和关联质量，从而对两个组成部分提供直观的评估。它通过提供更平衡的评估，解决了旧度量标准的局限性，适合于长期跟踪评估。

B. 实现细节

公共和私人检测。为了评估 FLASH 的跟踪能力，我们将其应用于 MOT17 和 MOT20 的公共和私人数据集，以及 DanceTrack 的私人数据集和 BDD100K 的验证集。在 MOT17 的公共数据集中，我们利用了 CenterTrack 的精细检测 [80], [81], [82], [45] ，而对于 MOT20 的公共数据集，则使用了 Tracktor 的精细检测 [88], [89], [81], [45] 。另一方面，对于 MOT17 和 MOT20 的私人数据集、DanceTrack 的测试集和 BDD100K 的验证集，我们应用了 YOLO-X 检测 [90] ，考虑了 ByteTracker 的训练流程 [43] 。架构。我们的 SMART-OD 流水线集成了三个最先进的框架：Segment Anything Model 2 (SAM2) [32] ，用于高质量实例分割，YOLO-World [17] ，用于高效的开放词汇检测，以及 Slicing Aided Hyper Inference (SAHI) [33] ，用于稳健的验证。这一多阶段架构将具有不同光照条件的输入图像转化为统一的掩模表示，支持在整个数据集中进行一致的目标检测。至于我们的 FLASH 框架，在线目标关联模块利用了 ByteTracker [43] ，并优化了参数：跟踪阈值为 0.6，匹配阈值为 0.7，注释与分割处理程序 (ASH) 采用了 SAM2 [32] 作为其基于内存的分割骨干。我们将结合 SAM2 和 ASH 的这种特定配置称为 SAM2ASH。这一零样本系统无需在目标数据集上进行训练，并通过复杂的 IoU 基匹配来维持目标身份。为实现内存效率，FLASH 实施了子集帧处理和基于批次的目标传播（每批次 5-10 个目标），支持任意长度序列的处理且内存使用受限。系统会根据资源可用性和序列复杂性在全序列和基于块的处理（50 帧块，重叠 10 帧）之间自动切换。

C. FLASH 作为多目标跟踪器

有关详细的实施细节，包括参数配置和实验设置，请参阅附录 D 。

1) 消融研究：FLASH 框架性能的组件分析： FLASH 框架整合了几个关键模块，这些模块协同工作以提供高质量的视频实例分割。为了理解每个组件的贡献，我们进行了一项消融研究，分析去除或修改特定模块如何影响整个系统性能。初始化模块的重要性初始化模块对于稳健处理长视频序列至关重要。如果没有这个模块在处理循环中运行，系统故障（例如由于内存段的高参数设置引起的内存不足错误）将需要从头开始重启整个过程。此模块实施的检查点管理系统通过以下方式提供基本的恢复能力：

1) 在处理过程中于战略点维护序列化的系统状态
2) 在中断后从特定帧启用恢复功能
3) 实施确保数据完整性的三相检查点协议

强大的检查点机制在处理长视频序列或跟踪超出内存限制或需要中断工作流程的大量对象时提供了显著价值。通过允许从最后保存的状态继续，而不是强制完全重启，它极大地提高了大规模视频标注任务的效率。值得注意的是，如果方法从全序列处理切换到块模式，系统需要从序列的开始启动，因为全序列检查点无法适应块模式检查点。在线对象关联模块的关键作用移除在线对象关联模块将会引入两个重大问题：

1) 计算效率低下：如果没有这个模块来过滤连续帧之间的冗余对象，系统将会处理许多重复的对象，从而大大增加计算成本。这些冗余会生成大量多余的多边形段，这些段需要在后期处理中移除，进一步增加了计算开销。
2) 遮挡场景中的伪影生成：如图 5 所清晰展示，当物体部分或完全被遮挡时（如在时间点 T1 和 T2 所示），标准的多目标跟踪器即使对于被遮挡的物体仍然生成边界框。没有物体关联模块来保持时间一致性时，基于记忆的分割会产生不与实际物体正确对齐的不一致的伪影多边形。这导致在连续帧中相同物体的分割模式不同，严重降低整体性能。

ASH 后处理步骤的必要性 ASH 模块的后处理能力，特别是冗余段的合并，为分割结果提供了必要的精细化。如图 6 所示，τ_{merge} 参数的变化具有显著影响：

1) 使用 $\tau_{\text{merge}} = 0.3$ ，可以实现最佳的分段合并
2) 使用 $\tau_{\text{merge}} = 0.7$ ，消除了一些冗余
3) 在 $\tau_{\text{merge}} = 0.85$ 中，许多冗余段仍然存在

这突显出，即使在线关联参数为了最小化计算开销而有意设置得较为宽松，后处理步骤仍能有效地处理剩余的冗余。时间上的平滑和合并操作保证了视频序列中对象表示的一致性，解决了帧对帧处理过程中不可避免的缺陷。

这些消融实验共同展示了 FLASH 架构的每个组成部分如何为视频实例分割创建一个稳健、内存高效的系统，即使在有遮挡和计算资源有限的挑战性场景中，也能保持时间一致性。

2) FLASH 与最新多目标跟踪方法的比较： 在公共检测基准上的表现。我们的实验评估显示，与最先进的方法相比，FLASH 在 MOT17 和 MOT20 公共检测基准上表现出不同的性能特征。正如表 I 所示，FLASH 在 MOT17 公共检测上的表现不佳，HOTA 分数为 43.60 %，MOTA 为 34.3 %，显著低于 GHOST [45] （50.7 % HOTA, 61.6 % MOTA）和 ArTIST-C [81] （48.9 % HOTA, 62.3 % MOTA）等领先方法。这个性能差距可以归因于公共检测的固有限制，它们提供了较低质量的边界框，从而对 SAM2 的分割初始化产生不利影响。尽管整体表现有限，FLASH 在身份保留能力上表现显著，在 MOT17 上实现了第三少的身份切换次数为 1283，仅次于使用 Tracktor 检测的 GHOST（1144）和 TrackPool（1188）。这种优越的 IDSW 表现表明 SAM2 的基于记忆的有效跟踪机制，可以在成功初始化后保持物体身份的一致性。然而，较为一般的 IDF1 得分 56.9 % 表明这一身份一致性未能转化为高关联质量，这是由于从不良初始检测传播的错误正例。在 MOT20 公共检测上，FLASH 保持了竞争性关联表现，IDF1 得分为 52.0 %，超越了包括 SORT [93] （45.1 %）和 GMPHD [94] （43.5 %）在内的多个知名方法。然而，MOTA 表现为 24.2 % 仍然显著低于最佳水平，强调了整体跟踪性能对检测质量的关键依赖。

私有检测基准的性能表现。

在如表 II 所示的私人检测基准评估中，采用高质量检测器，当检测质量限制最小化时，为 FLASH 的基本功能提供了关键见解。在 MOT17 私人检测上，FLASH 实现了 43.4 % 的 HOTA 和 28.7 % 的 MOTA，大大落后于 SPAM [61] （67.5 % HOTA，80.7 % MOTA）和 SUSHI [49] （66.5 % HOTA，81.1 % MOTA）等领先方法。尽管检测质量得到了提高，这种持续的性能差距揭示了 FLASH 在处理拥挤的行人跟踪场景时，当检测部分被遮挡时，其方法固有的

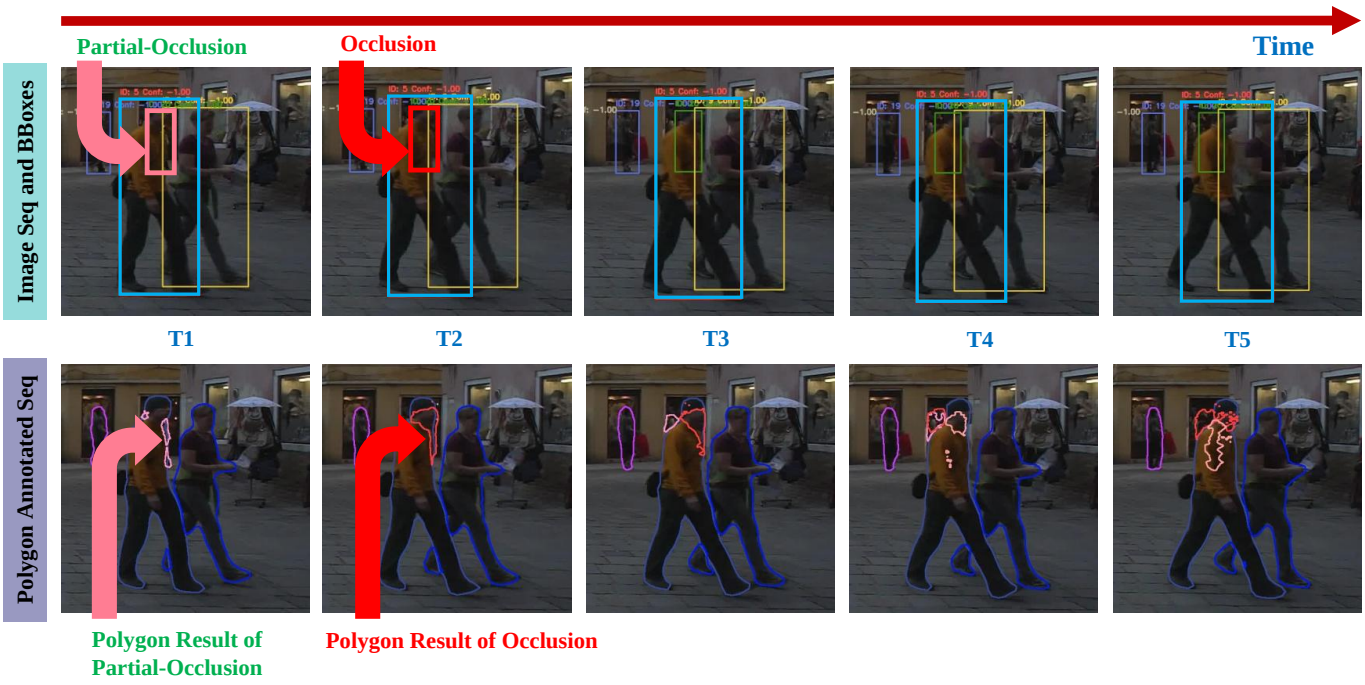

Fig. 5. 如果没有在线关联模块，当来自标准多目标跟踪中遮挡情况的边界框直接输入到 FLASH 时，不可避免地会产生伪影片段，导致整体性能下降。

Fig. 6. YOLO-World、SAM2-YW 和 SMART-OD 在 MOT17 训练集上的关键检测指标。SMART-OD 通过优化精确度-召回率权衡，达到了最高的精确度（0.728）和 MOTA（-0.014）。

局限性。

在 MOT20 私有检测数据上，FLASH 维持了一致但次优的表现，HOTA 为 38.6 %，MOTA 为 22.3 %，显著低于表现较好的方法，包括 SPAM [61]（65.8 % HOTA，76.5 % MOTA）、SUSHI [49]（64.3 % HOTA，74.3 % MOTA）、和 UTM（62.5 % HOTA，78.2 % MOTA）。在 MOT20 上的 IDF1 分数为 50.1 %，显示出相对较好的身份保留能力，但仍明显落后于 SPAM [61]（81.9 %）和 SUSHI [49]（79.8 %）。这种在 MOT17 和 MOT20 私有基准上的表现模式强化了 FLASH 在密集行人场景中面临的基本挑战，即检测和关联的精确性变得至关重要。

值得注意的是，FLASH 在 DanceTrack 数据集上表现出

TABLE I
在 MOT17 公共和 MOT20 公共检测上的 MOT 性能比较

Method	MOT17 Public				MOT20 Public			
	HOTA ↑	MOTA ↑	IDF1 ↑	IDSW ↓	HOTA ↑	MOTA ↑	IDF1 ↑	IDSW ↓
FLASH ‡	43.6	34.3	57.0	1283	39.8	24.2	52.0	2196
GHOST ‡ [45]	50.7	61.6	63.5	1715	-	-	-	-
GHOST † [45]	47.4	56.5	60.6	1144	43.4	52.7	55.3	1437
ArTIST-C ‡ [81]	48.9	62.3	59.7	2062	-	-	-	-
ArTIST † [81]	-	-	-	-	41.6	53.6	51.0	1531
CenterTrack ‡ [82]	48.2	61.5	59.6	3039	-	-	-	-
Tracktor v2 † [88]	44.8	56.3	55.1	1987	42.1	52.6	52.7	1648
TrackPool † [91]	-	55.9	60.5	1188	-	-	-	-
UNS † [92]	46.4	56.8	58.3	1914	-	-	-	-
SORT [93]	-	-	-	-	36.1	42.7	45.1	4470
GMPHD [94]	-	-	-	-	35.6	44.7	43.5	7492

第一名, 第二名, 第三名
Uses Tracktor refined detections
Uses CenterTrack refined detections

TABLE II
MOT 性能比较在 MOT17 PRIVATE, MOT20 PRIVATE, DANCETRACK 和 BDD100K 上

Method	MOT17 Private			MOT20 Private			DanceTrack			BDD100K					
	HOTA ↑	MOTA ↑	IDF1 ↑	HOTA ↑	MOTA ↑	IDF1 ↑	HOTA ↑	MOTA ↑	IDF1 ↑	mHOTA ↑	mMOTA ↑	mIDF1 ↑	HOTA ↑	MOTA ↑	IDF1 ↑
FLASH	43.4	28.7	56.5	38.6	22.3	50.1	62.0	64.1	72.5	53.7	-111.6	47.8	58.8	11.1	55.8
SPAM [61]	67.5	80.7	84.6	65.8	76.5	81.9	64.0	89.2	63.4	-	-	-	-	-	-
SUSHI [49]	66.5	81.1	83.1	64.3	74.3	79.8	63.3	88.7	63.4	-	-	-	-	-	-
GHOST [45]	62.8	78.7	77.1	61.2	73.7	75.2	56.7	91.3	57.7	45.7	44.9	55.6	61.7	68.1	70.9
ByteTrack [43]	62.8	78.9	77.1	60.4	74.2	74.5	47.7	89.6	53.9	45.4	45.2	54.6	61.6	68.7	70.2
MotionTrack [95]	65.1	81.1	80.1	62.8	78.0	76.5	-	-	-	-	-	-	-	-	-
UTM [96]	64.0	81.8	78.7	62.5	78.2	76.9	-	-	-	-	-	-	-	-	-
QDTrack [97]	63.5	78.7	77.5	60.0	74.7	73.8	54.2	87.7	50.4	41.7	36.3	51.5	60.9	63.7	71.4
MOTR [98]	57.8	68.6	73.4	-	-	-	54.2	79.7	51.5	-	32.0	43.5	-	-	-
FairMOT [99]	59.3	73.7	72.3	54.6	61.8	67.3	39.7	82.2	40.8	-	-	-	-	-	-
TrackFormer [41]	57.3	74.1	68.0	54.7	65.7	68.6	-	-	-	-	-	-	-	-	-
MeMOT [100]	56.9	72.5	69.0	54.1	66.1	63.7	-	-	-	-	-	-	-	-	-
TETer [101]	-	-	-	-	-	-	-	-	-	-	39.1	53.3	-	-	-
Yu et al. [34]	-	-	-	-	-	-	-	-	-	-	25.9	44.5	-	56.9	66.8

第一名, 第二名, 第三名

色，在所有评估方法中取得了最高的 IDF1 分数 72.5 %，并在 HOTA 中排名第二 (62.0 %)。这种出色的身份保持性能验证了 SAM2 在物体保持鲜明视觉特征的场景中基于记忆的追踪能力。优异的 IDF1 性能，结合具有竞争力的 HOTA 得分，展示了 FLASH 在跨时间序列维持长期身份一致性方面的特别优势。

在 BDD100K 验证集上，FLASH 实现了显著的类别平衡性能，具有最高的 mHOTA 得分 53.7 % ，这表明当所有物体类别被平等对待时，其跟踪质量优越。这一表现表明 SAM2 的时间一致性机制在各种物体类型中尤其有效，包括行人、车辆和骑自行车的人。

3) 理解 FLASH 的可变性能： FLASH 的性能在不同数据集上变化显著。虽然它在 DanceTrack 上取得了卓越的结果，但在 MOT17 上的表现不佳，而在 MOT20 和 BDD100K 上的表现则中等。尽管 FLASH 旨在保持一致的 ID 跟踪，但在各数据集上的不一致表现源于三个主要因素：

1) 由于初始边界框不佳导致的误报：不同指标上性能较低主要是由于高数量的误报造成的。这些误报源于提供的初始边界框质量差，当这些边界框被给予 SAM2 时，不可避免地会导致跟踪错误的对象而非目标类别。这个问题在边界框提供给部分或完全被遮挡的对象时尤为突出，会导致跟踪前面对象的一部分（而不是遮挡的对象）的伪影或甚至将两个对象分配给一个轨迹。这一问题在这样的边界框未被过滤掉而通过在线关联时尤其存在。
2) 遮挡处理中的差异：在 MOT17 和 MOT20 数据集中，为被遮挡的物体提供了真实值。然而，由于 FLASH 将 SAM2 的多边形转换为边界框，当物体完全被遮挡时，它无法提供任何边界框，或者对于部分遮挡的物体转换后的边界框与 MOT 挑战中的真实边界框不一致。即使物体实际上被正确跟踪，这种差异也显著降低了评价指标。
3) SAM2 的架构限制：当具有相似外观的物体紧密互动或互相遮挡时，SAM2 基于记忆的跟踪无法有效区分它们，导致身份混淆和跟踪失败。这一限制是 SAM2 架构固有的，因为它严重依赖于外观和空间线索，而当相似外观的物体互动时，这些线索变得模糊不清。

D. 时间段检测扫描 (SAM2Auto)

将 SMART-OD 与 FLASH 跟踪器集成，展示了我们完整的 SAM2Auto 流程的有效性，它通过基于记忆的持久性将仅检测结果转换为一个稳健的多目标跟踪系统。这一功能使得该流程成为自动标注应用程序的可靠引擎。

在我们对 SMART-OD 检测性能的消融研究（详细内容见附录 D）之后，我们将 FLASH 应用于 SMART-OD 在 MOT17 训练集上检测到的最终目标，以评估完整 SAM2Auto 流程的有效性。表 III 展示了全面的性能比较，突出显示在多个评估指标上的显著改进。

1) 基于内存的追踪：核心创新：SAM2Auto 的核心优势在于其基于内存的追踪机制：一旦 SMART-OD 首次检测到一个对象，它将在整个序列中持续被检测和追踪。这消除了在后续帧中重新检测的需要，并从根本上将追踪从逐帧检测转变为持续的对象维护。从-0.014 到 0.181 的大幅 MOTA 提升（提高了 1393 %）直接展示了这一能力。尽管 SMART-OD 的高精度（72.8 %）提供了质量初始检测，FLASH 的 SAM2 内存在后续视觉挑战中确保这些对象保持追踪。52.2 % 的召回率提升反映了基于内存的追踪如何恢复会在单个帧中被遗漏的对象。
2) 战略设计验证：我们的精确度为中心的检测策略被证明对基于记忆的跟踪是最优的。由于记忆系统的持久跟踪能力，对误报（22.6 % ）和漏报（31.2 % ）的控制性增加得到显著抵消。DetA 的提升为 206 %，展示了基于记忆的连续性如何将整体系统性能提升到超越单帧检测的程度。完整的 SAM2Auto 管道通过两阶段的方法实现其设计目标：SMART-OD 提供高精度的初始检测，而 FLASH 的 SAM2 记忆确保了随后的持久跟踪。这提供了可靠的开放词汇多目标跟踪，无需序列特定的优化，使其在多样化的现实场景中具有实际应用价值，在初始检测后连续的目标跟踪是至关重要的。

TABLE III
SMART-OD 检测与完整 SAM2AUTO 管道在 MOT17 训练集上的性能对比

Metric	SMART-OD	SAM2Auto	Improvement
MOTA	-0.014	0.181	+0.195 (1,393 %)
Precision	0.728	0.828	+0.100 (13.7 %)
Recall	0.224	0.341	+0.117 (52.2 %)
False Positives	2,516	3,084 *	+568 (22.6 %)
False Negatives	24,206	31,770 *	+7,564 (31.2 %)
DetA	0.084	0.257	+0.173 (206 %)
IDF1	N/A	1.156	New metric
HOTA	N/A	0.406	New metric

* Average per sequence values from tracking results

1) 区分 SAM2Auto 与 SPAMming： 由于在自动标注方面，SPAMming [61] 是唯一最接近我们方法的，因此在比较结果之前，重要的是要强调 SAM2Auto 和 SPAMming 之间的基本差异，如表 IV 所总结的。这些方法上的差异直接影响跟踪性能，并解释了我们实验结果的变化。为了公平比较，我们仅考虑它们在最少标注工作量 (3.3 %) 下的性能。我们还考虑了当提供标签时，FLASH 作为 SAM2Auto 的特殊情况的结果。

标注策略。最显著的区别在于标注方法。虽然 SPAMming 利用 3.3 % 的人工标注真实值数据来训练和微调其两个组件（标注引擎和追踪器），但 SAM2Auto 完全在无标注的情况下运行，仅依赖于 SMART-OD 进行自动目标检测。这种零标注限制虽然消除了人工标注成本，但也固有地限制了该方法适应数据集特定特征的能力，而这些特征是人工标注能够提供的。在 FLASH 的情况下，由于以前的跟踪研究提供了精心整理的标注，模型在使用 100 % 的真实值标注训练集进行训练时采用这些标注，但这些标注不包括用于评估的测试集。

TABLE IV
注释方法的比较标准

Method	Label Type	Label Effort	Tracker	Training Required
SAM2Auto	SMART-OD (Auto)	0 %	SAM2ASH	No
FLASH	Curated	0 %	SAM2ASH	No
SPAMming [61]	SPAM	3.3 %	ByteTrack	Yes
		3.3 %	GHOST	Yes

训练与适应。SPAMming 通过微调其标注管道和跟踪组件（ByteTrack 或 GHOST），利用可用的真实标签，使系统能够适应目标数据集的特定物体外观和运动模式。相比之下，SAM2Auto 使用预训练的 SMART-OD 和 SAM2ASH 跟踪器，没有进行任何针对于数据集的特定训练，使其更具普适性，但在特定场景下可能不如 SPAMming 优化。

追踪架构。追踪组件在设计理念上也存在根本性的差异。SPAMming 使用传统的在线 MOT 追踪器（ByteTrack/GHOST），即使在遮挡期间，通过运动模型和外观特征来维持物体预测。此外，这些追踪器通过低和高置信度阈值进行微调，以根据置信度分和追踪片段的寿命来保留或删除追踪片段。然而，SAM2ASH 是一个基于记忆的追踪器，它基于视觉外观生成遮罩片段，并且在完全遮挡期间不提供预测，这会影响到有遮挡物体的真值数据集中的追踪指标。更准确地说，虽然 GHOST 被描述为一个在线追踪器，但它实际上通过使用完整序列来调整其重新识别权重而使用离线关联。相比之下，SAM2ASH 作为一个离线追踪器操作，同时在线对象关联，其中使用 ByteTrack 来减少整体追踪的计算开销，从而避免处理冗余对象。值得注意的是，SAM2ASH 中的 ByteTrack 参数被配置为不根据置信度分过滤掉任何对象，因为我们期待从标注组件收到高质量的对象。

这些架构和方法上的差异为了解我们实验评估中观察到的性能权衡提供了背景，其中 SAM2Auto 的完全自动化特性以某些跟踪精度为代价，而相比于利用人工注释的方法。

2) SAM2Auto 结果： 实验结果提供了自动化标注系统中跟踪能力与检测质量之间性能权衡的全面视角。通过比较 SAM2Auto、FLASH 和 SPAMming，我们可以隔离每个组件的影响，并了解实现完全自动化跟踪的路径。

FLASH 是 SAM2Auto 的上限。FLASH 的性能展示了 SAM2Auto 在提供高质量精选标签时的跟踪能力。在 MOT17 上，FLASH 达到了 43.4 的 HOTA，28.7 的 MOTA 和 56.5 的 IDF1，显示即使有良好的标签，与 SPAMming 的经过微调的跟踪器相比，性能仍有差距（HOTA：51.6，MOTA：64.0，以及 IDF1：63 %）。这种差距可以归因于 SPAMming 在 3.3 % 的真实数据上进行的特定数据集训练，使得其跟踪器可以适应每个数据集的特定运动模式和外观特征。此对比揭示了尽管 SAM2ASH 是一个有能力的跟踪器，数据集特定的适应性使 SPAMming 享有大约 8-10 个

TABLE V
自动注释性能在 MOT17 私有、MOT20 私有、DANCETRACK 和 BDD100K 上的比较

Method	MOT17 Private			MOT20 Private			DanceTrack			BDD100K					
	HOTA ↑	MOTA ↑	IDF1 ↑	HOTA ↑	MOTA ↑	IDF1 ↑	HOTA ↑	MOTA ↑	IDF1 ↑	mHOTA ↑	mMOTA ↑	mIDF1 ↑	HOTA ↑	MOTA ↑	IDF1 ↑
SAM2Auto	40.5	10.3	50.0	32.3	11.6	40.8	37.4	-92.6	36.7	38.9	-271.6	32.6	56.6	8.3	51.8
FLASH	43.4	28.7	56.5	38.6	22.3	50.1	62.0	64.1	72.5	53.7	-111.6	47.8	58.8	11.1	55.8
SPAM-GHOST [61]	51.3	61.9	62.1	47.0	58.2	60.7	41.0	76.3	44.8	-	-	-	-	-	-
SPAM-ByteTrack [61]	51.6	64.0	63.0	47.9	57.6	61.4	39.5	76.4	45.0	-	-	-	-	-	-

第一名, 第二名, 第三名

HOTA 点的优势。

SAM2Auto: 测量检测影响。SAM2Auto 和 FLASH 之间的性能差异直接量化了自动检测质量的影响。在 MOT17 上，尽管 HOTA 从 43.4 下降到 40.5（3 点）的差距较小，但 MOTA 的下降更为显著，从 28.7 下降到 10.3（18.4 点）。这表明，当 SMART-OD 的自动检测引入较少的假阳性时，相比于人工校准的标注，通过时间跟踪这些假阳性以及漏检可以加剧指标的下降。要实现与 SPAMming 性能匹配的全自动跟踪，检测组件需要在 MOT17 上改进约 11 HOTA 点（从 40.5 到 51.6）。在 MOT20 上也呈现出类似的模式，SAM2Auto 需要从 32.3 提高到 47.9 HOTA 以匹配 SPAMming。

DanceTrack：SAM2ASH 的真正潜力。DanceTrack 提供了有力的证据，表明当参考标准与其设计相符时，SAM2ASH 的能力。FLASH 达到了卓越的性能（HOTA: 62.0, MOTA: 64.1, IDF1: 72.5），大幅优于 SPAMming（HOTA: 39.5-41.0）。这一戏剧性的逆转表明，当跟踪协议没有因为完全被遮挡的物体预测缺失而进行惩罚时，SAM2ASH 表现出色。MOT Challenge 的参考标准包括部分和完全被遮挡物体的注释，这与 SAM2ASH 的基于记忆的设计存在不匹配，因为它仅提供对可见物体的跟踪。DanceTrack 的协议更关注可见舞者，更好地与 SAM2ASH 的优势对齐，揭示了其真正的跟踪潜力。

BDD100K: 多类挑战。BDD100K 的结果进一步展示了检测质量的差距。虽然单类 HOTA 得分相对接近（SAM2Auto: 56.6 vs FLASH: 58.8），但多类 mMOTA 值对于两种方法都是严重负数（分别是-271.6 和-111.6）。这表明类似于 MOT Challenge 数据集，部分遮挡的真实框对 SAM@ASH 结果造成了惩罚。此外，自动检测在多类场景中特别困难，尽管 FLASH 的较好 mMOTA 表明即使在复杂的驾驶场景中，精心标注的标签显著减少了误报。

IV. 通向全自动化标注的路径

结果清楚地描绘了实现与半监督方法相媲美的完全自动化注释的路线图。虽然 SMART-OD 的消融研究 D 证明了检测流程的有效性，但将 SAM2Auto 与 FLASH 进行比较表明，检测质量仍然是主要瓶颈，而不是跟踪能力。该发现为自动跟踪系统的未来改进提供了明确方向。关键是，当与强跟踪结合时，甚至少量的误报也会通过序列传播，严重降低整体性能。

更具体地说，错误正例抑制成为自动标注环境中检测的主要挑战。虽然检测文献传统上强调召回率和精确度指标，但辨别真实目标与虚假检测并有效过滤这些错误候选的问题仍然缺乏研究。这个差距对于跟踪应用程序尤其关键，因为每一个错误正例都可能生成一个错误的轨迹，并在整个序列中持续存在。

SAM2Auto 管道也面临一个固有的限制：只有当 SMART-OD 首次检测到对象时，才能对其进行标注，这就造成了对象出现与初始检测之间的标注空白。虽然通过改进 SMART-OD 来使其在场景进入时立即检测对象可以解决该问题，但考虑到诸如遮挡和不利的光照条件等实际限制，这种完美的检测是不现实的。

一种更实用的解决方案涉及双向跟踪：通过从初始检测点进行前向和后向跟踪，我们可以恢复完整的对象轨迹。这种方法显著降低了检测要求，而不是要求帧完美的初始检测，整个流程只需要在整个序列中对每个对象检测一次。这种单次检测要求更容易实现，并且可以在保持系统完全自动化的性质的同时实现全面的注释覆盖。

FLASH 的性能分析，特别是在 DanceTrack 上的优秀结果，揭示了关于 SAM2ASH 的能力和局限性的重要见解。当跟踪器在可见度清晰和物体交互适中的序列中表现出色时，面对频繁遮挡和视觉上相似的物体时性能显著下降。我们识别出三个具体挑战并提出相应的解决方案：

1) 视觉相似性的架构增强：SAM2 的当前架构在区分视觉相似对象时存在困难，特别是在遮挡发生时。结合额外的判别特征，例如运动模式、时间一致性或学习的对象特定嵌入，可以显著提高在拥挤场景中的身份保留能力。
2) 自适应段合并：尽管将 τ_{merge} 设置得相对较低，但在复杂场景中冗余段仍然存在。一个根据场景复杂性和对象密度调整的自适应阈值可以消除冗余，同时保留不同的对象，通过减少误报直接提高 MOTA 分数。
3) 误检轨迹抑制：当前的流程假设 SMART-OD 可以实现完美检测，需要手动移除误检轨迹——这种方法无法扩展。为实用性部署，引入自动误检抑制，可能通过基于置信度的过滤或轨迹分析是必要的。

总体而言，这些增强功能——双向跟踪、视觉相似性处理和假阳性抑制——形成了一条明确的途径，能够在保持零人工标注这一关键优势的同时，实现媲美半监督方法的全自动跟踪。

在这篇论文中，我们提出了 SAM2Auto，这是一个完全自动化的视频标注流程，消除了多目标跟踪中手动标注的需要。通过结合 SMART-OD 进行自动目标检测和 FLASH 进行基于记忆的跟踪，我们证明了在多样化数据集上实现零标注跟踪的可行性。关键的是，我们的整个流程在目标数据集上不需要进行任何训练或微调，仅依赖于预训练模型，却仍然实现了具有竞争力的性能。我们还介绍了 FLASH，它通过提供高质量的精心策划标签，将视频目标分割和多目标跟踪连接起来，展示了在与精确检测结合时我们跟踪组件的最高性能。

我们的全面评估隔离了检测质量与跟踪能力的影响，揭示了检测——特别是误报抑制——仍然是主要的瓶

颈。尽管 SAM2Auto 取得了可观的性能，但与半监督方法的差距是明确且可以解决的。值得注意的是，FLASH 在 DanceTrack 上的卓越表现表明，在适当的评估协议下，我们的跟踪架构甚至可以超越经过微调的方法。这些发现强调，有竞争力的全自动跟踪不仅是可行的，而且可能只需要在检测质量、双向跟踪和视觉相似性架构增强方面进行相对温和的改进。

SAM2Auto 代表了视频标注民主化的重要一步，让大型跟踪对没有资源进行广泛人工标注或计算训练的研究人员和从业者变得可及。随着检测模型的不断改进和提出的增强技术的实施，我们设想，完全自动无训练的跟踪将不仅成为一种具有成本效益的替代方案，还将成为许多现实世界应用的首选方法。视频理解的未来在于能够无需人工干预就能学习和注释的系统，而这项工作为实现这一目标提供了一个实用的框架和清晰的路线图。

我们诚挚地感谢加拿大数字研究联盟（前身为 Compute Canada）通过 NSERC 联盟计划提供的计算资源支持。

REFERENCES

[1] E. Ramírez, "I was surprised by the great response to my recent post on small object detection—thank you for all the dms and shares." LinkedIn post, August 2024. Accessed: August 29, 2024.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2021.

[4] A. Rocky, Q. J. Wu, and W. Zhang, "Review of accident detection methods using dashcam videos for autonomous driving vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 8356–8374, 2024.

[5] R. Girshick, "Fast r-cnn," 2015.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[8] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[9] R. Bommasani *et al.*, "On the opportunities and risks of foundation models," 2022.

[10] Z. Yang, Z. Li, M. Shao, D. Shi, Z. Yuan, and C. Yuan, "Masked generative distillation," 2022.

[11] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your vit? data, augmentation, and regularization in vision transformers," 2022.

[12] A. D'Alessandro, A. Mahdavi-Amiri, and G. Hamarneh, "Afreeca: Annotation-free counting for all," 2024.

[13] Y. Lu, C. Xu, X. Wei, X. Xie, M. Tomizuka, K. Keutzer, and S. Zhang, "Open-vocabulary point-cloud object detection without 3d annotation," 2023.

[14] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, and Y. Wang, "Emernerf: Emergent spatial-temporal scene decomposition via self-supervision," 2023.

[15] L. Jin, R. Tucker, Z. Li, D. Fouhey, N. Snavely, and A. Holynski, "Stereo4d: Learning how things move in 3d from internet stereo videos," 2024.

[16] O. Zohar, A. Lozano, S. Goel, S. Yeung, and K.-C. Wang, "Open world object detection in the era of foundation models," in *arXiv preprint arXiv:2312.05745*, 2023.

[17] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," 2024.

[18] R. Khirodkar, T. Bagautdinov, J. Martinez, S. Zhaoen, A. James, P. Selednik, S. Anderson, and S. Saito, "Sapiens: Foundation for human vision models," 2024.

[19] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," 2024.

[20] T. Ren, Y. Chen, Q. Jiang, Z. Zeng, Y. Xiong, W. Liu, Z. Ma, J. Shen, Y. Gao, X. Jiang, X. Chen, Z. Song, Y. Zhang, H. Huang, H. Gao, S. Liu, H. Zhang, F. Li, K. Yu, and L. Zhang, "Dino-x: A unified vision model for open-world object detection and understanding," 2024.

[21] H. Song and J. Bang, "Prompt-guided detr with roi-pruned masked attention for open-vocabulary object detection," *Pattern Recognition*, vol. 155, p. 110648, 2024.

[22] L. Yao, J. Han, Y. Wen, X. Liang, D. Xu, W. Zhang, Z. Li, C. Xu, and H. Xu, "Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection," 2022.

[23] S. Wu, W. Zhang, S. Jin, W. Liu, and C. C. Loy, "Aligning bag of regions for open-vocabulary object detection," 2023.

[24] L. Yao, J. Han, X. Liang, D. Xu, W. Zhang, Z. Li, and H. Xu, "Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment," 2023.

[25] X. Zuo, P. Samangouei, Y. Zhou, Y. Di, and M. Li, "Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding," 2024.

[26] J. Wu, Y. Jiang, Q. Liu, Z. Yuan, X. Bai, and S. Bai, "General object foundation model for images and videos at scale," 2023.

[27] D. Kim, A. Angelova, and W. Kuo, "Region-aware pretraining for open-vocabulary object detection with vision transformers," 2023.

[28] S. Xing, C. Qian, Y. Wang, H. Hua, K. Tian, Y. Zhou, and Z. Tu, "Openemma: Open-source multimodal model for end-to-end autonomous driving," 2024.

[29] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby, "Simple open-vocabulary object detection with vision transformers," 2022.

[30] S. Zhao, Z. Zhang, S. Schulter, L. Zhao, V. K. B. G, A. Stathopoulos, M. Chandraker, and D. Metaxas, "Exploiting unlabeled data with vision and language models for object detection," 2022.

[31] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," 2024.

[32] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[33] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 966–970, 2022.

[34] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," 2020.

[35] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," 2020.

[36] H. Zhang, L. Xiao, X. Cao, and H. Foroosh, "Multiple adverse weather conditions adaptation for object detection via causal intervention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 3, pp. 1742–1756, 2024.

[37] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," 2020.

[38] Z. Cui, K. Li, L. Gu, S. Su, P. Gao, Z. Jiang, Y. Qiao, and T. Harada, "You only need 90k parameters to adapt light: A light weight transformer for image enhancement and exposure correction," 2022.

[39] M. Afifi, K. G. Derpanis, B. Ommer, and M. S. Brown, "Learning multi-scale photo exposure correction," 2021.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2020.

[41] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[42] V. D. Stanojevic and B. T. Todorovic, "Boosttrack: boosting the similarity measure and detection confidence for improved multiple object tracking," *Machine Vision and Applications*, vol. 35, p. 53, April 2024.

[43] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking byăassociating every detection box," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow,

M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 1–21, Springer Nature Switzerland, 2022.
[44] W. Lv, Y. Huang, N. Zhang, R.-S. Lin, M. Han, and D. Zeng, "Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19321–19330, 2024.
[45] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, and L. Leal-Taixé, "Simple cues lead to a strong multi-object tracker," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13813–13823, 2023.
[46] W. Li, B. Li, J. Wang, W. Meng, J. Zhang, and X. Zhang, "Romot: Referring-expression-comprehension open-set multi-object tracking," *The Visual Computer*, June 2024.
[47] M. Segu, L. Piccinelli, S. Li, Y.-H. Yang, B. Schiele, and L. V. Gool, "Samba: Synchronized set-of-sequences modeling for multiple object tracking," 2024.
[48] Z. Liu, X. Wang, C. Wang, W. Liu, and X. Bai, "Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.
[49] O. Cetintas, G. Brasó, and L. Leal-Taixé, "Unifying short and long-term tracking with graph hierarchies," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22877–22887, June 2023.
[50] S. Li, L. Ke, M. Danelljan, L. Piccinelli, M. Segu, L. Van Gool, and F. Yu, "Matching anything by segmenting anything," *CVPR*, 2024.
[51] J. Zhang, Y. Cui, G. Wu, and L. Wang, "Joint modeling of feature, correspondence, and a compressed memory for video object segmentation," 2023.
[52] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing, "Putting the object back into video object segmentation," 2024.
[53] R. Goyal, W.-C. Fan, M. Siam, and L. Sigal, "Tam-vt: Transformation-aware multi-scale video transformer for segmentation and tracking," 2024.
[54] N. Homayounfar, J. Liang, W.-C. Ma, and R. Urtasun, "Videoclick: Video object segmentation with a single click," 2021.
[55] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking anything with decoupled video segmentation," in *ICCV*, 2023.
[56] F. Raji, L. Ke, Y.-W. Tai, C.-K. Tang, M. Danelljan, and F. Yu, "Segment anything meets point tracking," 2023.
[57] J. Qi, Y. Gao, Y. Hu, X. Wang, X. Liu, X. Bai, S. Belongie, A. Yuille, P. H. S. Torr, and S. Bai, "Occluded video instance segmentation: A benchmark," 2022.
[58] H. Wang, C. Yan, S. Wang, X. Jiang, X. Tang, Y. Hu, W. Xie, and E. Gavves, "Towards open-vocabulary video instance segmentation," 2023.
[59] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker3: Simpler and better point tracking by pseudo-labelling real videos," in *Proc. arXiv:2410.11831*, 2024.
[60] J. Wang, Z. Wu, D. Chen, C. Luo, X. Dai, L. Yuan, and Y.-G. Jiang, "Omnitracker: Unifying visual object tracking by tracking-with-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–15, 2025.
[61] O. Cetintas, T. Meinhardt, G. BrasÃş, and L. Leal-TaixÃł', "Spamming labels: Efficient annotations for the trackers of tomorrow," in *European Conference on Computer Vision (ECCV)*, 2024.
[62] A. Osep, T. Meinhardt, F. Ferroni, N. Peri, D. Ramanan, and L. Leal-TaixÃł', "Better call sal: Towards learning to segment anything in lidar," in *European Conference on Computer Vision (ECCV)*, 2024.
[63] Y. Xiong, C. Zhou, X. Xiang, L. Wu, *et al.*, "Efficient track anything," *preprint arXiv:2411.18933*, 2024.
[64] A. Bagchi, Z. Bao, Y.-X. Wang, P. Tokmakov, and M. Hebert, "Refereverything: Towards segmenting everything we can speak of in videos," 2024.
[65] C.-Y. Yang, H.-W. Huang, W. Chai, Z. Jiang, and J.-N. Hwang, "Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory," 2024.
[66] A. Alimohammadi, S. Nag, S. A. Taghanaki, A. Tagliasacchi, G. Hamarneh, and A. M. Amiri, "Smite: Segment me in time," 2024.
[67] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3460–3469, 2015.
[68] I. Elezi, Z. Yu, A. Anandkumar, L. Leal-Taixe, and J. M. Alvarez, "Not all labels are equal: Rationalizing the labeling costs for training object detection," 2021.
[69] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," 2024.
[70] L. Yao, R. Pi, J. Han, X. Liang, H. Xu, W. Zhang, Z. Li, and D. Xu, "Detclipv3: Towards versatile generative open-vocabulary object detection," 2024.
[71] Q. Ma, S. Yang, L. Zhang, Q. Lan, D. Yang, H. Chen, and Y. Tan, "Apovis: Automated pixel-level open-vocabulary instance segmentation through integration of pre-trained vision-language models and foundational segmentation models," *Image and Vision Computing*, vol. 154, p. 105384, 2025.
[72] B. Yan, M. Sundermeyer, D. J. Tan, H. Lu, and F. Tombari, "Towards real-time open-vocabulary video instance segmentation," 2024.
[73] M. Liang, J.-C. Su, S. Schulter, S. Garg, S. Zhao, Y. Wu, and M. Chandraker, "Aide: An automatic data engine for object detection in autonomous driving," 2024.
[74] A. J. Yang, S. C. Romero, M. Dvornik, S. Segal, *et al.*, "Automatic labeling of objects from lidar point clouds via trajectory-level refinement," December 2024.
[75] N. Agarwal *et al.*, "Cosmos world foundation model platform for physical ai," 2025.
[76] J. Witt, "Launch: Auto label images with roboflow," *Roboflow Blog*, March 2024.
[77] S. S. Mullappilly, A. S. Gehlot, R. M. Anwer, F. Shahbaz Khan, and H. Cholakkal, "Semi-supervised open-world object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, p. 4305–4314, Mar. 2024.
[78] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231, AAAI Press, 1996.
[79] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," 2016.
[80] S. Karthik, A. Prabhu, and V. Gandhi, "Simple unsupervised multi-object tracking," *CoRR*, vol. abs/2006.02609, 2020.
[81] F. S. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and inpainting for multiple object tracking," *CoRR*, vol. abs/2012.02337, 2020.
[82] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," 2020.
[83] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," 2020.
[84] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[85] R. Kasturi, D. B. Goldgof, P. Soundararajan, V. Manohar, J. S. Garofolo, R. Bowers, M. Boonstra, V. N. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 319–336, 2009.
[86] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set forămulti-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops* (G. Hua and H. Jégou, eds.), (Cham), pp. 17–35, Springer International Publishing, 2016.
[87] J. Luiten, A. Ošep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, vol. 129, p. 548–578, Oct. 2020.
[88] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
[89] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[90] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," *CoRR*, vol. abs/2107.08430, 2021.
[91] C. Kim, L. Fuxin, M. Alotaibi, and J. M. Rehg, "Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9548–9557, 2021.
[92] F. Bastani, S. He, and S. Madden, "Self-supervised multi-object tracking with cross-input consistency," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang,

and J. W. Vaughan, eds.), vol. 34, pp. 13695–13706, Curran Associates, Inc., 2021.
[93] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sept. 2016.
[94] N. L. Baisa, "Online multi-object visual tracking using a gm-phd filter with deep appearance learning," in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2019.
[95] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "Motiontrack: Learning robust short-term and long-term motions for multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17939–17948, June 2023.
[96] S. You, H. Yao, B.-k. Bao, and C. Xu, "Utm: A unified multiple object tracking model with identity-aware feature enhancement," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21876–21886, 2023.
[97] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 164–173, June 2021.
[98] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking withătransformer," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 659–675, Springer Nature Switzerland, 2022.
[99] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.
[100] J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu, and S. Soatto, "Memot: Multi-object tracking with memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8090–8100, June 2022.
[101] S. Li, M. Danelljan, H. Ding, T. E. Huang, and F. Yu, "Tracking every thing inătheăwild," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 498–515, Springer Nature Switzerland, 2022.

APPENDIX

A. SMART-OD 数学公式

本节提供了在第 II-A 节中描述的 SMART-OD 流水线的完整数学公式。三阶段流水线包括分割、掩码引导分析和鲁棒阈值化，每个阶段都有特定的数学定义和参数配置。

1) 分割阶段： 第一阶段使用 SAM2 对场景中所有潜在物体进行自动掩膜生成。分割过程定义为：

$$I_{\text{Masked}} = \text{AutoMask}_{\text{SAM2}}(I, \theta_s, \theta_o, \theta_n) \tag{21}$$

，其中 I_{Masked} 表示生成的实例掩码，$\text{AutoMask}_{\text{SAM2}}$ 是 SAM2 分割函数，I 是输入图像，θ_s 是稳定性得分阈值，θ_o 是稳定性得分偏移，θ_n 是框的非极大值抑制 (NMS) 阈值参数。

2) 掩膜引导检测阶段： 第二阶段使用 YOLO-World 对掩码表示进行开放词汇对象检测。检测过程被表述为：

$$D_{\text{init}} = \text{OVOD}(I_{\text{Masked}}, C, \theta_c, \theta_i, \theta_n) \tag{22}$$

，其中 D_{init} 表示初始检测结果，OVOD 是 YOLO-World 开放词汇表检测函数，I_{Masked} 是方程 21 中的掩码标注图像，C 是自定义目标类别集合，θ_c 是置信度阈值，θ_i 是 IoU 阈值，θ_n 是 NMS 阈值参数。

为了消除不合理大小的检测，应用面积比率过滤：

$$D_{\text{filtered}} = \{d \in D_{\text{init}} \mid \theta_{\min} < \frac{A_d}{A_I} < \theta_{\max}\} \tag{23}$$

其中 A_d 是检测区域，d ，A_I 是总图像区域，而 $\theta_{\min}$ 和 $\theta_{\max}$ 分别是最小和最大面积比阈值。

最后，算法 2 给出了 SMART-OD 用于处理数据集每个序列的综合算法：

Algorithm 2 SMART-OD：目标检测和验证流程

Require: Image $\mathcal{I}$, parameters Θ , thresholds δ
Ensure: Verified detections $\mathcal{V}_{\text{box}}$, $\mathcal{V}_{\text{class}}$, $\mathcal{V}_{\text{score}}$

// Stage 1: Segmentation
$\mathcal{M} \leftarrow \text{AutoMask}_{\text{SAM2}}(\mathcal{I}, \Theta_s, \Theta_o, \Theta_n)$ ▷ Generate masks
// Stage 2: Mask-guided Analysis
$\mathcal{D}, \mathcal{L} \leftarrow \text{OVOD}(\mathcal{M}, \mathcal{C}, \Theta_c, \Theta_i, \Theta_n)$ ▷ Detect objects
// Stage 3: Robust Verification
$\mathcal{C} \leftarrow \text{DBSCAN}(\mathcal{D}_{\text{xyxy}}, \epsilon, \mu)$ ▷ Cluster detections
$\mathcal{R} \leftarrow \{[\min_{b \in \mathcal{C}_i} x_1, \min_{b \in \mathcal{C}_i} y_1, \max_{b \in \mathcal{C}_i} x_2, \max_{b \in \mathcal{C}_i} y_2] \mid \mathcal{C}_i \in \mathcal{C}\}$
$\theta_d \leftarrow \text{DynamicThreshold}(\mathcal{D}_{\text{conf}})$ ▷ Adaptive threshold
$\theta_{\text{final}} \leftarrow \max(\theta_d, \theta_{\min})$
$\mathcal{V}_{\text{idx}}, \mathcal{V}_{\text{class}}, \mathcal{V}_{\text{box}}, \mathcal{V}_{\text{score}} \leftarrow \emptyset$
for $r \in \mathcal{R}$ **do**
  $\mathcal{P} \leftarrow \text{SAHI}(r, \mathcal{M})$ ▷ Process ROI
  for $d \in \mathcal{D}$ where $d \subset r$ **do**
    if $\max_{p \in \mathcal{P}} \text{IoU}(d, p) > \theta_v$ and $c_d > \theta_{\text{final}}$ **then**
      $\mathcal{V}_{\text{idx}} \leftarrow \mathcal{V}_{\text{idx}} \cup \{d_{\text{idx}}\}$
      $\mathcal{V}_{\text{class}} \leftarrow \mathcal{V}_{\text{class}} \cup \{d_{\text{class}}\}$
      $\mathcal{V}_{\text{box}} \leftarrow \mathcal{V}_{\text{box}} \cup \{d_{\text{box}}\}$
      $\mathcal{V}_{\text{score}} \leftarrow \mathcal{V}_{\text{score}} \cup \{d_{\text{score}}\}$
    end if
  end for
end for
return $\mathcal{V}_{\text{box}}, \mathcal{V}_{\text{class}}, \mathcal{V}_{\text{score}}$

B. FLASH 实现细节

1) 初始化模块： 该模块实施了一个稳健的三相检查点管理，以防止在保存操作期间的数据丢失：

1) 写入到临时文件：$f_{temp} \leftarrow \text{serialize}(\mathcal{X})$
2) 创建现有检查点 $f_{backup} \leftarrow f_{checkpoint}$ 的备份（如果存在）
3) 将临时文件提升为检查点：$f_{checkpoint} \leftarrow f_{temp}$

其中 $\mathcal{X}$ 表示被检查点的数据（即系统的当前状态），serialize($\mathcal{X}$) 将这些数据转换为可以保存在磁盘上的格式。这种三阶段方法确保即使系统在保存过程中崩溃，仍然至少有一个有效的检查点文件。检查点包括多边形和边界框视频段、处理进度和跟踪信息。

此外，该模块的恢复功能通过从检查点文件名中提取帧号，以根据保存的检查点确定正确的起始点，如下所示：

根据这种恢复逻辑，当加载检查点时，系统从其文件名中提取元数据以决定处理应该从哪里恢复：如果标记为“initial”，处理从头开始（由特殊值-1 表示）；如果标记为“final”，处理从最后一个完全处理的帧（v_{frames}）中的最大帧号）继续；否则，如果检查点的文件名中包含特定的帧号 N，则处理恰好从该帧恢复。这种稳健的机制确保了在中断情况下处理的连续性，这在处理由于计算或时间限制而无法在单一会话中完成的长视频序列时至关重要。

2) 在线对象关联模块：

a)： 时间上下文处理：系统根据检测的时间背景进行不同的处理。在第一帧中，所有有效的检测都被初始化为需要分割的新物体。在后续帧中，检测首先使用 IoU 阈值与现有轨迹匹配，只有那些没有匹配的才被认为是新物体。这种方法确保了适当的初始化，并在整个视频序列中保持连续性。

对于所有帧，该模块首先使用稳健的验证标准提取有效的边界框：

$$\text{ValidBox}(B, W, H) = \begin{cases} \text{true} & \text{if size}(B) \in [\lambda_{min}, \lambda_{max}] \ \wedge \\ & \text{boundaries}(B) \subset [m, W-m] \times \\ & [m, H-m] \ \wedge \\ & \text{aspect}(B) \in [0.2, 5.0] \\ \text{false} & \text{otherwise} \end{cases} \tag{24}$$

，其中 λ_{min} 和 λ_{max} 定义了允许的尺寸范围，W 和 H 表示框架尺寸，m 是边距参数，而 $\text{aspect}(B)$ 计算框的纵横比。

b)： 检测-轨迹关联：本模块的核心功能是通过帧与帧之间的检测到轨迹映射来建立时间一致性。此过程通过系统地将新的检测与现有的轨迹关联起来，以在视频序列中保持对象身份。

形式上，该系统维护一个映射 $\mathcal{M} : D_j \rightarrow \mathcal{T}$ ，将每个检测索引分配给一个轨迹 ID：

$$\mathcal{M}(i) = \begin{cases} t_k & \text{if } \exists k : \text{IoU}(B_{\mathcal{T}_k}, B_{d_i}) > \tau \\ & \text{and } k = \arg\max_l \text{IoU}(B_{\mathcal{T}_l}, B_{d_i}) \end{cases} \tag{25}$$

对于帧 j 中的每个检测 $d_i \in D_j$ ，系统识别出具有最高 IoU 重叠的轨迹：

$$\text{matched\_track\_idx} = \arg\max_{k} \text{IoU}(B_{\mathcal{T}_k}, B_{d_i}) \tag{26}$$

其中，$B_{\mathcal{T}_k}$ 表示轨迹 $\mathcal{T}_k$ 的边界框，B_{d_i} 是检测 d_i 的边界框。当最大 IoU 超过阈值 τ （通常为 0.5）时，匹配被确认：

$$\text{IoU}(B_{\mathcal{T}_{\text{matched\_track\_idx}}}, B_{d_i}) > \tau \tag{27}$$

一旦确认，检测 d_i 被关联到轨迹 $\mathcal{T}_{\text{matched\_track\_idx}}$ ，并且将相应的轨迹 ID 分配给该检测。

3) 标注与分割处理器 (ASH)： 为了实现有效的实时性能，ASH 实施了两个关键的内存优化策略：

子集帧处理：对于一个包含帧 $\mathcal{F} = \{F_1, F_2, ..., F_T\}$ 的视频序列，ASH 在帧 t 处使用帧子集 $\mathcal{S}$ ，定义为：$\mathcal{S}_t = \{F_t, F_{t+1}, ..., F_T\}$ ，其中 T 是序列中的总帧数，t 是在线目标关联模块启动新对象的帧。这使得系统可以将检测到新对象的帧的掩码传播到序列的结束，避免过多的内存需求。

批量对象处理：为了优化计算效率，ASH 以批量方式处理检测到的对象。在帧 t 检测到的新对象集 $\mathcal{N}_t = \{o_1^t, o_2^t, ..., o_n^t\}$ 被划分为批次：

$$\mathcal{B}_k = \{o^t_{(k-1)\cdot\beta+1}, o^t_{(k-1)\cdot\beta+2}, ..., o^t_{\min(k\cdot\beta, n)}\} \tag{28}$$

其中，n 是在时间 t 的新对象数，β 是批量大小参数，和 $k \in \{1, 2, ..., \lceil n/\beta \rceil\}$ 。这种方法在吞吐量和内存约束之间取得了平衡，这在处理有限 GPU 硬件上具有众多对象的帧时尤为重要。批量大小参数（ β ）控制同时处理的对象数量，平衡内存需求和计算效率。

因此，对于每个对象批次，基于内存的分割模型会使用适当的后续帧进行初始化。这种状态管理策略确保每个批次都能以一个干净的模型状态开始，同时保持适当的视频上下文。

长视频序列的基于块的处理流程中，FLASH 和 FLASH 的整体过程在算法 3 和算法 4 中详述。

Algorithm 3 FLASH：视频对象分割的注释和分割处理器

Require: New objects $\mathcal{N}_t$ at frame t , video sequence $\mathcal{F} = \{F_1, F_2, ..., F_T\}$, tracking data, segmentation model ϕ , batch size $\boldsymbol{\beta}$, detection-track IoU threshold $\boldsymbol{\tau}_{\text{track-det}}$, redundant segment merge threshold $\boldsymbol{\tau}_{\text{merge}}$, temporal smoothing factor $\boldsymbol{\alpha}$, mask content threshold $\boldsymbol{\epsilon}$

Ensure: Video segments (polygons) $\mathcal{V}$

$\mathcal{V} \leftarrow \emptyset$ ▷ Initialize output segments
$\mathcal{S}_t \leftarrow \{F_t, F_{t+1}, ..., F_T\}$ ▷ Process only from current frame to end
for each batch $\mathcal{B}_k$ of up to β objects from $\mathcal{N}_t$ **do**
  // Object ID Assignment via Tracking
  for each $o_i^t \in \mathcal{B}_k$ **do**
    if $\exists j : \text{IoU}(b_i^t, b_j^{t-1}) > \tau_{\text{track}}$ **then**
      $\text{ID}(o_i^t) \leftarrow \text{track}_j$ ▷ Assign existing track ID
    else
      $\text{ID}(o_i^t) \leftarrow \max(\text{IDs}) + 1$ ▷ Assign new track ID
    end if
  end for
  // Mask Propagation
  for each $o_i^t \in \mathcal{B}_k$ **do**
    for $\delta \leftarrow 0$ **to** $(T - t)$ **do**
      $M_i^{t+\delta} \leftarrow \phi(o_i^t, F_t, F_{t+\delta})$ ▷ Generate masks
      $P_i^{t+\delta} \leftarrow \psi(M_i^{t+\delta})$ ▷ Convert to polygons
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{(t+\delta, \text{ID}(o_i^t), P_i^{t+\delta})\}$
    end for
  end for
end for
// Post-Processing
1. Remove Empty Masks
for each object o_i **do**
  $\tau(o_i) \leftarrow \max\{t \in [1, T] : \sum_{x,y} M_i^t(x, y) > \epsilon\}$
  Remove $(t, \text{ID}(o_i), P_i^t)$ from $\mathcal{V}$ for all $t > \tau(o_i)$
end for
2. Apply Temporal Smoothing
for each object o_i **do**
  for $t \leftarrow 2$ **to** $\tau(o_i)$ **do**
    $P_i^t \leftarrow \alpha \cdot P_i^t + (1 - \alpha) \cdot P_i^{t-1}$
    Update $(t, \text{ID}(o_i), P_i^t)$ to $(t, \text{ID}(o_i), P_i^t)$ in $\mathcal{V}$
  end for
end for
3. Merge Redundant Segments
for each frame $t \in [1, T]$ **do**
  for each pair (P_i^t, P_j^t) where $i \neq j$ in frame t **do**
    if $\text{IoU}(P_i^t, P_j^t) > \tau_{\text{merge}}$ **then**
      Merge $(t, \text{ID}(o_i), P_i^t)$ and $(t, \text{ID}(o_j), P_j^t)$ in $\mathcal{V}$
    end if
  end for
end for
return $\mathcal{V}$

C. SAM2Auto 实现细节

1) 代表序列选择： 第一步涉及识别数据集中具备最高目标密度的序列。这个“最坏情况下”的序列提供了一个参数优化的挑战性测试案例。我们选择整个数据集中具有最大对象数量的序列：

$$S_{rep} = \arg\max_{S \in \mathcal{D}} \max_{f \in S} |O_f| \tag{29}$$

其中，S 代表数据集 $\mathcal{D}$ 中的一个序列，$|O_f|$ 是帧 f 中的对象数量，$\max_{f \in S} |O_f|$ 确定了在每个序列中具有最高对象数量的帧。这个选择标准确保我们选择包含整个数据集中单个最拥挤帧的序列。

Algorithm 4 FLASH：用于长视频序列的稳健块处理

Require: Video sequence $\mathscr{F} = \{F_1, F_2, ..., F_T\}$, annotations $\mathscr{A}$, chunk size $\boldsymbol{\chi}$, overlap size $\boldsymbol{\omega}$
Ensure: Complete video segments (polygons) $\mathscr{V}$

$\mathscr{V} \leftarrow \emptyset$ ▷ Initialize output segments
$\mathscr{B} \leftarrow \emptyset$ ▷ Initialize bbox segments
// First attempt full processing
success ← TryFullProcessing($\mathscr{F}$, $\mathscr{A}$, $\mathscr{V}$, $\mathscr{B}$)
if not success **then**
  // Fall back to chunk-based processing
  $n \leftarrow \lceil T/(\chi - \omega) \rceil$ ▷ Number of chunks
  current_frame ← 1
  for $i \leftarrow 1$ **to** n **do**
    // Find optimal chunk boundaries
    optimal_frame ← FindOptimalFrame(current_frame, ω , $\mathscr{B}$)
    chunk_start ← max(1, optimal_frame − ω)
    chunk_end ← min(T, chunk_start + χ − 1)
    // Determine overlap region
    if $i > 1$ **then**
      overlap_frames ← [chunk_start, chunk_start + ω]
    else
      overlap_frames ← $\emptyset$
    end if
    // Process current chunk
    $\mathscr{V}_i, \mathscr{B}_i$ ← ProcessChunk($\mathscr{F}$, $\mathscr{A}$, chunk_start, chunk_end, overlap_frames)
    // Merge with previous results at overlap
    if $i > 1$ **then**
      $\mathscr{V}$ ← MergeOverlappingSegments($\mathscr{V}$, $\mathscr{V}_i$, overlap_frames)
      $\mathscr{B}$ ← MergeOverlappingBBoxes($\mathscr{B}$, $\mathscr{B}_i$, overlap_frames)
    else
      $\mathscr{V} \leftarrow \mathscr{V}_i$
      $\mathscr{B} \leftarrow \mathscr{B}_i$
    end if
    current_frame ← chunk_end +1
  end for
  // Apply final post-processing to full sequence
  $\mathscr{V}$ ← PostProcessPolygonSegments($\mathscr{V}$)
  $\mathscr{V}$ ← MergeRedundantSegments($\mathscr{V}$)
end if
return $\mathscr{V}$

我们将 SMART-OD 应用于选定序列中最拥挤的帧。优化过程旨在最大化对真实物体的检测，同时最小化误报。形式上，我们优化参数集 $\Theta = \{\theta_s, \theta_o, \theta_n, \theta_c, \theta_i, \theta_v\}$ 以最大化目标函数：

$$J(\Theta) = \alpha \cdot \text{Recall}(\Theta) + (1 - \alpha) \cdot \text{Precision}(\Theta) \tag{30}$$

其中，α 是一个加权因子，根据应用需求平衡召回率和精确率。

然后将优化参数应用于整个代表序列。使用标准指标评估性能：

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{31}$$

其中 TP、FP 和 FN 分别代表真正例、假正例和假负例。调整参数以确保两个指标都超过特定应用的阈值。

为了验证泛化性，我们将配置应用于从数据集的不同部分随机选择的序列 S_{val} 。如果满足条件，则该配置被认为是有效的：

$$\min(\text{Precision}(S_{\text{val}}), \text{Recall}(S_{\text{val}})) \geq \gamma \cdot \min(\text{Precision}(S_{\text{rep}}), \text{Recall}(S_{\text{rep}})) \tag{32}$$

其中 $\gamma \leq 1$ 是一个容差因子（通常是 0.9）。

验证后的 SMART-OD 配置使用分布式处理在整个数据集上部署。对于每个序列 S_i ，流水线生成：

$$D_i = \text{SMART-OD}(S_i, \Theta_{opt}) \tag{33}$$

其中 D_i 表示序列 S_i 的检测结果，而 Θ_{opt} 是优化的参数集。

*2) 全规模序列管理器应用：*序列管理器，以 FLASH 作为其核心处理组件，应用于检测结果：

$$V_i = \text{SequenceManager}(S_i, D_i, \Theta_{SM}) \tag{34}$$

其中 V_i 表示序列 S_i 的视频片段（多边形），Θ_{SM} 是控制 FLASH 行为和分块策略的序列管理器参数集。这种集成方法能够在所有序列中自动处理内存高效的处理，而无需人工干预。

*3) 质量保证：*最终注释在一个分层样本的序列上进行验证。对于每个抽样的序列，我们计算自动生成的注释与一小组手动注释之间的交并比（IoU）：

$$\text{IoU}(V_i, M_i) = \frac{|V_i \cap M_i|}{|V_i \cup M_i|} \tag{35}$$

序列中 IoU 低于阈值 τ_{QA} 的将进行针对性的参数优化。

D. 实验细节

*1) 实验装置：*在 SMART-OD 中，我们仔细配置了 SAM2 的自动掩膜生成器，优化了参数以平衡召回率和精度：稳定性分数阈值为 0.90，稳定性分数偏移为 0.7，以及框 NMS 阈值为 0.7。通过广泛的实验确定了这些参数，以确保在各种场景中产生稳健的掩膜。对于目标检测，我们配置了 YOLO-World 以识别自动驾驶场景中的八个特定类用于 BDD（行人、骑车人、汽车、卡车、公共汽车、火车、摩托车、自行车），并在 MOT17、MOT20 和 DanceTrack 数据集中仅识别人物类。为了在所有数据集中保持一致性能，我们采用了以下统一参数：YOLO 置信阈值为 0.001，YOLO IoU 阈值为 0.1，YOLO NMS 阈值为 0.1，验证 IoU 阈值为 0.03，最小面积比为 0.0008，最大面积比为 0.20。我们的动态阈值方法包括 kmeans、标准偏差的均值、标准偏差的 kmeans（默认）、以及双 kmeans。在 SMART-OD 的检测聚类阶段（3.1），我们将点之间的最大距离（ϵ ）设置为 100，将聚类中的最小样本数（μ ）设置为 1。该流程生成标准化的 MOT 格式输出。

对于 SAM2ASH，我们的参数优化集中于在处理效率与分割质量之间取得平衡。关键参数包括：批量大小（$\beta = 5$ ），控制同时处理的对象数量；块大小（$chi = 50$ ）帧，具有重叠大小（$omega = 10$ ）帧，以实现高效的序列处理；IoU 追踪阈值（$\tau_{\text{track-det}} = 0.5$ ），用于确定在在线跟踪过程中新检测

何时与现有的轨迹关联，同时使用一个单独的重叠阈值（$\tau_{\text{overlap}} = 0.7$ ）来合并处理块之间重叠帧中的片段；

和框验证约束符合方法部分中建立的标准：大小界限为 $\lambda_{\min} = 10$ 像素和 $\lambda_{\max} = 1000$ 像素，边界距离框架边缘 $m = 0.5$ 像素的外缘，以及 0.2 到 5.0 之间的宽高比约束，以确保只处理格式良好的边界框。我们将 ByteTracker 配置为具有跟踪阈值（$\tau_{\text{conf}} = 0.6$ ）、匹配阈值（$\tau_{\text{match}} = 0.7$ ）以及 20 帧的跟踪缓冲区，以实现稳健的 ID 维护。为确保时间一致性，我们应用了自适应平滑，基础因子（$\alpha = 0.2$ ）根据物体运动速度动态调整，减少抖动同时保留自然运动。我们采用 $\tau_{\text{merge}} = 0.3$ 的 IoU 阈值来识别并合并重叠的物体实例，使用最小遮罩内容阈值 $\epsilon = 3$ 像素来确定有效的遮罩表示，用于空遮罩移除过程。在将 FLASH 评估为具有精炼公共和私人检测的独立跟踪器时，我们将所有边界框的置信得分设置为固定值 0.95，以确保所有检测都能通过 ByteTracker 的在线关联，且没有被过滤掉。话虽如此，当将 FLASH 用作 SAM2Auto 的集成组件时，由于来自 YOLO-World 的检测对象的置信得分通常很低，我们将其置信得分通过线性重新缩放映射到 0.7 到 0.95 的范围，从而导致与 FLASH 的跟踪模式相同的条件：

$$\text{conf}_{\text{new}} = 0.7 + (\text{conf}_{\text{orig}} - \min_{\text{conf}}) \cdot \frac{0.25}{\max_{\text{conf}} - \min_{\text{conf}}} \quad (36)$$

。这确保了在不同检测来源和实施场景中的一致跟踪条件，同时允许 ByteTracker 的在线关联基于适当的置信度做出跟踪决策。

如 II-A 中所述，SMART-OD 流水线旨在为整个数据集中提供最多的真实正样本对象，而不需要针对每个帧或数据序列调整参数。为此，我们通过在 MOT17 [79] 训练集上的消融研究评估了每个组件的影响。如图 7 和表 VI 所示，我们比较了以下几种方案的性能：(i) YOLO-World 基线，(ii) 结合了 YOLO-World 的 SAM2（SAM2-YW），以及（iii）具有 SAHI 验证的完整 SMART-OD 流水线。

需要注意的是，在此检测阶段，对象在各帧之间不被赋予持久的身份，这导致了像 MOTA 这样以跟踪为导向的指标值较低。负的 MOTA 值（-0.154 到-0.014）反映了这一局限性，因为该指标对身份切换和跟踪不一致的惩罚非常严重。然而，我们的主要目标是评估检测性能，因此精确率和召回率才是更相关的指标。

结果表明，整个流程成功实现了其主要目标，即最大化真实阳性检测，同时将误报降至最低。最显著的改进来自误报的显著减少——从 YOLO-World 基准中的 5,861 减少到完整流程中的 2,516，减少了 57 %（表 VI ）。这种减少验证了我们采用强大的验证阶段来过滤掉虚假的检测结果而无需特定序列参数调整的方法。正如图 7 所示，流程在精度和召回率之间表现出战略性权衡。虽然召回率从 38.0 % 下降到 22.4 %，但精度显著提高，从 59.4 % 提高到 72.8 %。这种权衡在 SMART-OD 系统中是精心设计的，因为该系统优先考虑检测质量而不是数量。从-0.154 到-0.014 的多目标跟踪准确性（MOTA）提高进一步确认了这一权衡导致了更好的整体检测性能，因为 MOTA 既惩罚误报也惩罚漏报。

检测精度（DETA）指标从 0.128（YOLO-World）下降到 0.084（SMART-OD）。鉴于这个流程设计更重视精确度而非召回率，这种 DETA 的减少是预料之中的。尽管较低的 DETA 可能显得违反直觉，但这准确地反映了我们为了得到更高的检测置信度而接受较少检测数量的策略。MOTA 的显著改善，以及假阳性的大幅减少，证实了这一权衡为我们预定的应用带来了更优的检测质量。

值得注意的是，仅使用 SAM2 分割的中间配置（SAM2-YW）在基线之上几乎没有改善，精度和召回率的增益可以忽略不计（图 7 ），而实际上将假阳性增加到 6,564（表 VI ）。这一发现强调了单独使用分割是不够的；只有与 SAHI 验证阶段结合时，SAM2 的真正价值才能体现。整个流程的有效性源于 SAM2 提供的面罩引导分析与 SAHI 中稳健的阈值机制之间的协同交互，这种机制适应每个序列而无需人工干预。这些结果从实验上证明了 SMART-OD 成功为开放词汇检测提供了数据集优选的解决方案。通过接受受控的召回率降低，同时显著提高精度（图 7 ）并实现 57% 假阳性减少（表 VI ），该流程创造出一种更可靠的检测系统，其能够在不同序列间普遍化，而无需每个序列的参数优化，从而实现其设计目标，即高效的数据集级对象检测。

TABLE VI
SMART-OD 组件在 MOT17 训练集上的误报和漏报计数

Method	False Positives	False Negatives
YOLO-World	5,861	19,411
SAM2-YW	6,564	19,074
SMART-OD	2,516	24,206

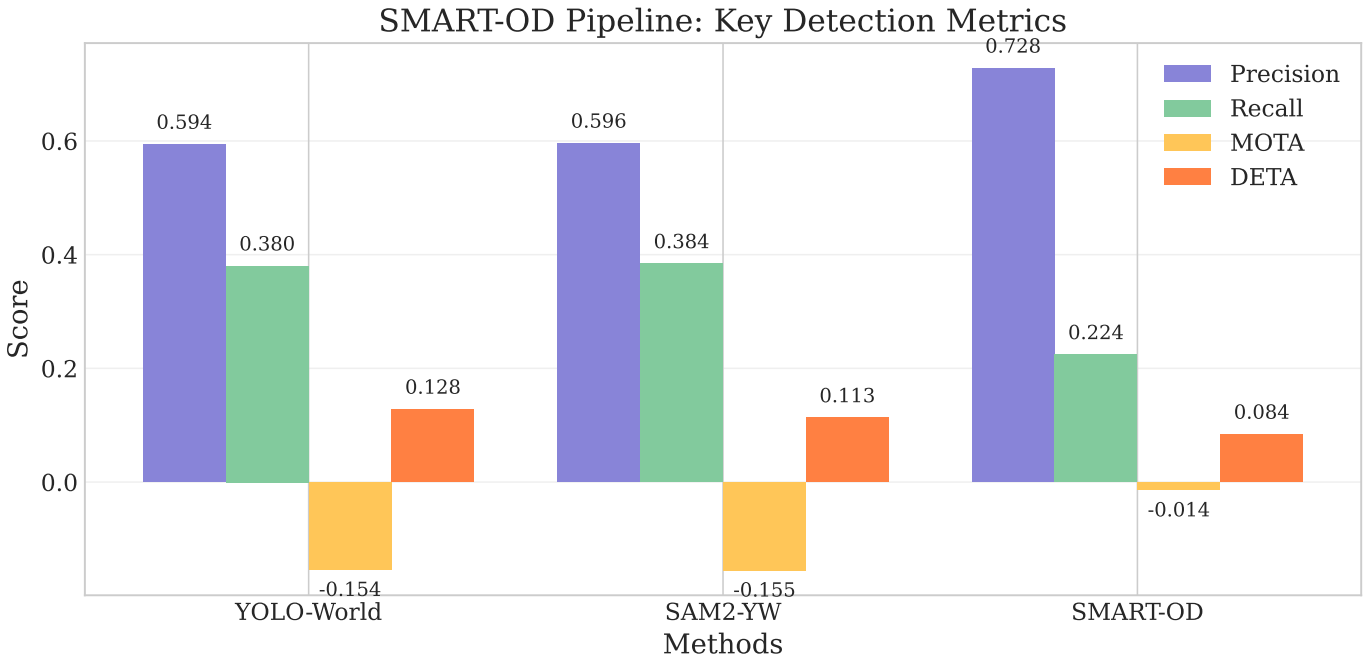

Fig. 7. 基于 MOT17 训练集的 YOLO-World、SAM2-YW 和 SMART-OD 的关键检测指标。SMART-OD 通过优化精度-召回率的权衡，实现了最高的精度（0.728）和 MOTA（-0.014）。