Abstract

身份切换仍然是多对象跟踪（MOT）算法必须处理的主要困难之一。 现在，许多最先进的方法都使用序列模型来解决此问题，但是其训练可能会受到降低效率的曝光偏差的影响。 在本文中，我们**介绍了一种新的训练过程，**该过程使算法面临其自身的错误，同时明确**尝试最小化 switches**，从而带来更好的训练。

我们提出了一种迭代方案，即构建一个丰富的训练集，并使用它来学习评分函数，该函数是目标跟踪指标的明确表示。无论是仅使用简单的几何特征还是要考虑外观的更复杂的几何特征，我们的方法在多个 MOT 基准上均优于最新技术。

1. Introduction

MOT 方法中的一个普遍关注的问题是防止身份切换，即将与不同目标相对应的轨迹错误地合并为一个对象。在拥挤的场景中很难做到这一点，尤其是当单个目标对象的外观不够鲜明时。许多最新的方法依赖于小轨迹（短轨迹段）而不是单个检测来跟踪目标对象。小轨迹可以合并为更长的轨迹，当发生身份切换时可以将其再次拆分。

大多数最先进的方法都依赖于深层网络，通常如[34、18、25、56、27]在此类轨迹上采用运行 RNN 架构形式。 这需要训练序列模型，并且要解决两个众所周知的问题中的一个或两个，我们的方法可以克服这些问题：

• **Loss-evaluation mismatch**. 当通过优化与推理过程中实际所需性能不符的度量进行训练时，会发生这种情况。 在 MOT 中，一个示例是使用分类损失来创建针对特定于跟踪的度量（例如 MOTA [6]或 IDF [46]）最佳的轨迹。**为了消除这种不匹配，我们引入了一种对小轨迹进行评分的原始方法，该方法是 IDF 指标的显式表示，可以在没有 GT 的情况下进行计算。我们使用它来确定对人的跟踪有多自信，预测更紧密的边界框位置以及估计实际轨迹是否超出了观察到的小轨迹。**

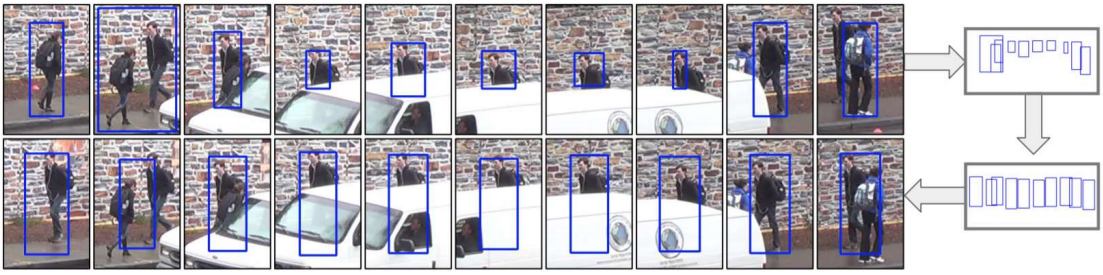


图 1.在困难情况下保持跟踪。第一行：由于过往车辆造成的遮挡，跟踪器可以轻松返回包含多个 id-switches 的轨迹。相机视野内的相应边界框显示在右侧。最底行：我们的算法不仅消除了身份切换，而且回归到一组更为严格的边界框。 在此示例中，我们的算法仅基于简单的几何特征进行了此操作，而无需使用外观信息。

• **Exposure bias**. 它源于模型在训练过程中没有暴露于自身的错误，并导致训练和推理/跟踪过程中观察到的数据分布非常不同。我们通过引入一种更为详尽，但在计算上可行的方法来训练模型时利用数据来消除这种偏差。**为此，在训练过程中，我们不限于仅使用检测到一两个人的小轨迹来进行训练[40、35、48]。取而代之的是，我们将跟踪算法产生的任何小轨迹分组视为一个潜在轨迹，通过共享许多常见检测来控制小轨迹的数量来防止组合爆炸**。这产生了更丰富的训练数据集，解决了曝光偏差问题，并使我们的算法能够处理令人困惑的情况，在这种情况下，跟踪算法可能会轻易地从一个人切换到另一个人，或者完全错过某个人。图 1 描绘了一种这样的情况。请注意，即使没有外观信息也可以执行此操作。

因此，我们的贡献是对这两个问题的解决。通过将其集成到仅使用非常简单的特征（边界框，检测器置信度）的算法中，我们的性能优于其他也不使用外观特征的方法。即使是都利用基于外观的特征，我们同样能胜过最先进的方法。总而言之，这些结果证明了我们培训

程序的有效性。

在本文的其余部分中，我们首先简要回顾了相关的工作以及减轻损失评估失配和曝光偏差的当前方法。然后，我们介绍我们的跟踪方法；它是多种假设跟踪的一种变体，旨在学习有效地对小轨迹评分。接下来，我们描述得分函数的确切形式及其减少不匹配和偏差。最后，我们介绍我们的结果。

2. Related work

多目标跟踪（MOT）具有悠久的历史，可追溯到雷达跟踪等应用[8]。随着对象检测器的最新改进，逐检测跟踪范式[2]已成为事实上的标准，并已证明对许多应用（例如监视或体育运动员跟踪）有效。它涉及到首先检测单个帧中的目标对象，将这些检测结果关联到称为轨迹小但可靠的短轨迹中，然后将这些轨迹连接成更长的轨迹。然后可以将它们用于解决诸如社交场景理解[1，3]，未来位置预测[31]或人类动态建模[15]之类的任务。

在将个体检测分组为轨迹时，很难保证每个结果轨迹代表单个个体的整个轨迹，即没有身份切换。

许多方法都依赖于外观[17、29、57、59、11、33、47]，动作[13]或社交提示[20、44]。它们主要用于关联检测对，并且仅考虑非常短期的相关性。但是，由于人们的运动轨迹通常在许多帧上都是可以预测的，**因此可以通过对较长时间段内的行为进行建模来获得优异的性能**[22、28、37]。带注释的训练数据和基准（例如 MOT15-17 [30，39]，DukeMTMC [46]，PathTrack [38]和 Wildtrack [9]）的可用性现在使学习利用这一知识所需的数据关联模型成为可能。由于这是我们的方法所要做的，因此我们在此简要回顾一些实现此目标的最新方法。

2.1. Modeling Longer Sequences

[43，42]的工作是使用 RNN 对长轨迹建模的最新近来方法之一。该算法估计 ground-plane 占用率，但不执行显式数据关联。[40]提出了一种通过预测目标的未来位置而无需使用外观特征即可执行数据关联的方法。随后采用了几种 MOT 方法，即使用序列模型使数据关联更加健壮，以便人们重新识别[48，35]，学习更好的社交模型[1]，预测未来位置[31，54]或联合检测，跟踪 ，以及活动识别[3]。

这些模型通常是根据与单个人的轨迹完全匹配或仅略微偏离其轨迹的样本轨迹进行训练的，从而使他们容易受到曝光偏差的影响。**此外，损失函数通常主要是为定位或识别而设计的，而不是针对 GT 轨迹的逼真度。这引入了与指标（通常为 IDF [46]或 MOTA [6]）的损失评估失配，从而更可靠地反映了算法的理想行为。**

大多数使用序列模型的最新方法都依赖于两种优化技术之一：**用于数据关联的分层聚类**[49、59、46、34、18、25]或**多个假设跟踪**[56、27、10] 。前者涉及没有共享假设的有效观察组，而后者则允许存在冲突的假设集，直到找到最终解决方案为止。与我们最相似的方法是[27]。它还使用多个假设跟踪器和序列模型的组合进行评分。但是，训练程序主要依赖于 GT 信息，因此更容易受到曝光偏差的影响。另一种密切相关的方法是[40]的方法，该方法仅从几何特征中训练序列模型进行数据关联，因此非常适合与我们的方法（仅使用几何线索）进行比较。这些方法都是最新的，共同代表了当前的最新技术。因此，在第 5 节中，我们将把它们作为基线，可以与我们的方法进行比较。

2.2 Reducing Bias and Loss-Evaluation Mismatch

由于暴露偏差和损失评估失配也是自然语言处理（NLP）[50]中的问题，尤其是机器翻译[53]中的问题，因此在这些领域中已经提出了几种方法来减少它[45，4]。但是，其中大多数操作是在这样的假设下进行的：输出序列可以包含预定义集中的任何字符。结果，他们通常依赖于波束搜索程序，该程序本身经常使用语言模型来生成包含正确候选者的各种候选集。更一般而言，允许训练模型在输入和输出之间没有可区分关系的技术（例如策略梯度[52]，直通估计[5]和 Gumbel-Softmax [23]）可以被视为减少暴露偏差的方法。

不幸的是，在 MOT 的情况下，这些检测形成了一个时空图，其中可以建立许多几乎相同的轨迹。这很容易使标准的波束搜索技术不堪重负：当将自己限制在得分最高的候选者以防止组合爆炸时，很容易发生：仅会考虑一组非常相似但虚假的轨迹，而忽略真正的轨迹。这种故障模式已在单对象跟踪和未来位置预测的背景下得到解决，[21，36]中采用了强化学习学习的跟踪策略，在[12]中通过对一批图像引入时空注意机制，从而确保批次内没有暴露偏差。取而代之的是，该算法依赖于已经获得的轨道的历史正样本，因此将其重新引入。对于 MOT，已经提出了一种基于强化学习的方法[55]来决定是创建新的跟踪还是终止旧的跟踪。在[48]中也解决了这个问题，但是序列模型的学习是独立完成的，并且仍然容易受到暴露偏差的影响。 [37]的方法尝试为 IDF 指标进行显式优化。它通过完善其他跟踪方法的输出来实现。这样可以减少损失评估失配，但是序列评分模型是硬编码的，而不是学习的，我们将证明学习它会产生更好的结果。

3. Tracklet-Based Tracking

我们的跟踪方法依赖于创建和合并 tracklets 以构建高得分轨迹，就像在多个假设跟踪中一样[26]。在本节中，我们将其形式化并描述其组成部分，前提是给出了评分函数。评分函数及其学习方法将在下一节中讨论。

3.1. Formalization

我们假设一个视频序列由 N 帧组成，我们分别对每一帧都运行行人检测算法。这会产生检测（行人）$d_i \in \mathbb{R}^4$的集合D，其中d_i的四个元素是图像中相应 bbox 的坐标信息。我们将 tracklet T 表示为$[d_1, d_2, \ldots, d_N]$。实际上，tracklets 很少会覆盖整个序列。对于行人定位未知的帧，我们将d_n设置为 0（即标记此帧目标丢失），所以，第一个非零列是其起点，最后一行是其终点。如果有两个 tracklets T_1和T_2都只包含相同的一个 detection，那么就将这两个 tracklets 合并成一个。

令Φ：$\mathbb{R}^{4\times N}\rightarrow\mathbb{R}^{F\times N}$是一个特征函数，它将维度$F$的特征向量分配给 tracklet 的每一列。实际上，这些特征可以是边界框坐标，置信度以及从前一帧中最近的检测偏移值。它们也可以是与检测相关联的基于图像的特征，我们将在 5.3 节中列出所有特征。让我们进一步假设，我们可以从这些特征计算出分数$S(\Phi(T))$，当 tracklet 真正代表一个人的轨迹时，该分数为高，否则为低。然后，可以将跟踪理解为建立使目标函数最大化的非重叠 tracklet T_j的集合，目标函数为：

$$\sum_j S(\Phi(\mathbf{T}_j)) \,. \qquad (1)$$

在本节的其余部分中，我们将假定函数S已经给定，并将低分数分配给可以生成的各种不良候选轨迹，将高分数分配给真实轨迹。

3.2. Creating and Merging Tracklets

我们迭代合并 tracklets 以创建包括真实轨迹的更长的候选轨迹，同时抑制许多候选轨迹以防止计算上不可行的组合爆炸。然后我们贪婪地选择一个最佳子集。当两个轨迹在有较大的重合区域，我们认为它们是重叠的。更具体地，如果由两个 tracklets 的 bbox 共享的像素总数（该值用每个 tracklet 中的 bbox 的面积的总和的最小值进行归一化）高于阈值C_{IOU}，则认为重叠。**我们还将消除小于 N（批次长度）或分数低于另一个阈值C_{score}的 tracklet。**C_{IOU}和C_{score}是我们在验证集上估计的超参数。概述的过程涉及以下两个主要步骤。

3.2.1 Generating Candidate Trajectories

候选轨迹的集合必须包含真实轨迹，但其大小必须保持足够小以防止组合爆炸。为此，

给定初始检测集D，我们将其作为初始跟踪集。然后，我们对 n 从 2 到 N 重复以下两个步骤：

1. Growing：合并可以合并的 tracklets 对，其结果将比两个 tracklets 最大的还大 1。具有k_1和k_2非零 detections 的 tracklets 产生一个具有$max(k_1, k_2)$+1 非零 detections 的 tracklet，其包含它们两个所有的非零 detections。
2. Pruning: 给定 tracklet T1，对于在 Growing 期间与它合并的所有 T2，仅保留使得分$S(\Phi(\cdot))$最大化的合并。

该过程使假设的数量相对于检测的数量保持线性。但是，它保留了所有可能检测到的候选对象。即使在修剪过程中尽早犯了错误，这也可以防止算法过早地失去人员并终止轨迹。我们在图 2（b）中给出一个示例。在附录中，我们将此启发式方法与其他几种方法进行了比较，并表明它在防止组合爆炸的同时又不丢失有效的假设是有效的。

3.2.2 Selecting Candidates

给定生成的 tracklets 集，我们想选择一个最大化我们目标函数的兼容子集。为此，我们选择一个假设的子集，该假设的子集应具有最佳的总分，并且要遵守非重叠约束。我们从得分最高的轨迹开始贪婪地做这件事。如附录中所述，我们还尝试了一种更为复杂的方法，将该方法强制转换为以最佳方式求解的整数程序，并且结果相似。

4. Learning to Score

等式 1 的得分函数$S(\Phi(\cdot))$是第 3 节跟踪过程的核心。当我们创建和合并 tracklet 时，我们希望它支持那些可以与单个人相关联而无需身份切换的跟踪，即在 IDF 指标方面得分较高的跟踪。关于评价指标我们选择了 IDF 而不是其他流行的替代方法，例如 MOTA，因为它已经显示出对 id-switches 更敏感[46]。

在本节的其余部分中，我们首先定义S，我们使用图 2（a）所示的深度网络来实现。然后我们描述我们如何训练它。

4.1. Defining the Scoring Function

理想情况下，对于每个 tracklet T 和对应的 GT 轨迹 G，我们都应该有$S(\Phi(T)) \approx IDF(T, G)$。但是，在推断时，G 的是未知的。为了克服这个困难，请回顾[46]， tracklet T = [d1，…。 。 。 ，dn]和 GT 轨迹 G = [g1，.. 。 。 ，gn]的 IDF 定义为与 GT 匹配的检测次数的两倍，超过两者的总长度之和：

$$\mathbf{IDF}(\mathbf{T}, \mathbf{G}) = \frac{2 \cdot \sum_{n:\mathbf{d}_n \neq \mathbf{0}, \mathbf{g}_n \neq \mathbf{0}} \mathbb{1}(IoU(\mathbf{d}_n, \mathbf{g}_n) > 0.5)}{|\{n : \mathbf{d}_n \neq \mathbf{0}\}| + |\{n : \mathbf{g}_n \neq \mathbf{0}\}|}, \quad (2)$$

where IoU is the intersection over union of the bounding boxes. To approximate it without knowing $\mathbf{G}$, we write

$$S(\Phi(\mathbf{T})) = \frac{2 \cdot \sum_{n:\mathbf{d}_n \neq \mathbf{0}, lab_n > 0.5} iou_n}{|\{n : \mathbf{d}_n \neq \mathbf{0}\}| + |\{n : lab_n > 0.5\}|}, \quad (3)$$

assuming that the deep network of Fig. 2, (a) has been trained to regress from $\mathbf{T}$ to

- iou_n: the prediction of intersection over union of the $\mathbf{d}_n$ and $\mathbf{g}_n$ boxes;
- lab_n: the prediction of whether the ground truth trajectory exists in frame n.

我们还训练我们的网络以预测对边界框d_n的必要更改，以产生 GT 边界框g_n，我们将其表示为sft_n。 它不用于计算S，但可以在推理过程中用于使观察到的边界框与 GT

更好地对齐。

为了训练网络以预测上面介绍的lab_n，iou_n和sft_n值，我们定义了一个损失函数，该损失函数是预测与 GT 之间的误差之和。 我们写成:

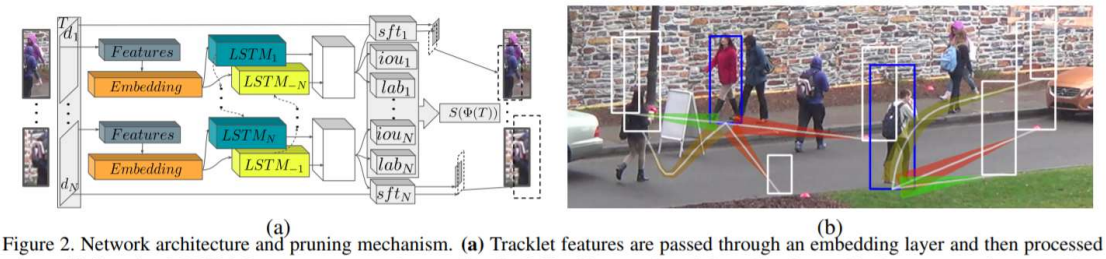
$$L(\mathcal{T},\mathcal{G}) = \sum_{n=1}^{N} L_{lab}(\mathbf{d}_n,\mathbf{g}_n) + \sum_{n:\mathbf{d}_n\neq\mathbf{0}} L_{iou}(\mathbf{d}_n,\mathbf{g}_n) + \sum_{n:\mathbf{d}_n\neq\mathbf{0}} L_{sft}(\mathbf{d}_n,\mathbf{g}_n), \quad (4)$$

$$L_{lab}(\mathbf{d}_n,\mathbf{g}_n) = ||lab_n - \mathbb{1}(\mathbf{g}_n \neq \mathbf{0})||_2,$$
$$L_{iou}(\mathbf{d}_n,\mathbf{g}_n) = ||iou_n - IoU(\mathbf{d}_n,\mathbf{g}_n)||_2,$$
$$L_{sft}(\mathbf{d}_n,\mathbf{g}_n) = 1 - IoU(\mathbf{d}_n + sft_n,\mathbf{g}_n),$$

where $\mathbf{d}_n + sft_n$ denotes the shifting the bounding box $\mathbf{d}_n$ by sft_n.

可以说，我们可以训练网络直接回归到 IDF，而不是先估计lab_n，iou_n和sft_n，然后使用等式 3 的近似值进行计算。但是，我们的实验表明，像我们所做的那样，在每个时间步上询问更详细的反馈都迫使网络更好地理解运动，而平均预测通常可以很好地估计 IDF。

我们选择不对损失函数的组成部分应用任何权重因子，因为它的组成部分可以看作是识别 FP（当 label 应为零时）和 FN（当 IoU <0.5 时），并且由于我们想权衡两者均等，我们没有对$\mathrm{L}_{lab}, \mathrm{L}_{sft}, \mathrm{L}_{iou}$使用任何权重因子。



(a) (b)

Figure 2. Network architecture and pruning mechanism. **(a)** Tracklet features are passed through an embedding layer and then processed using a bi-directional LSTM. Its outputs are used to predict the IoU with ground truth bounding boxes (iou), presence of a person in a scene (lab), and regress bounding box shift to obtain ground truth bounding boxes (sft). **(b)** Candidate tracklets starting from two different bounding boxes in blue and ending with bounding boxes in white. In this case, during pruning phase the best tracklets, shown in green, are assigned the highest score and retained, and all others are eliminated.

图 2.网络架构和修剪机制。（a）Tracklet 特征通过嵌入层，然后使用双向 LSTM 进行处理。它的输出被用来预测与 GT 边界框（iou）的 IOU，场景中是否有人的存在（lab）和回归边界框平移来获得 GT 边界框（sft）。（b）候选 tracklets，从两个不同的蓝色边框开始，以白色的边框结束。 在这种情况下，在修剪阶段，以绿色显示的最佳小轨迹将分配最高分数并保留，而所有其他小轨迹将被消除。

4.2. Training Procedure

训练网络时避免暴露偏差的关键是提供丰富的训练集。为此，我们在以下两个步骤之间进行交替：

1. 在评估 S 时，使用当前网络权重运行 3.2 节的假设生成算法；
2. 将新创建的 tracklets 添加到训练集中，并执行单个训练时期。

除了学习网络权重之外，此过程还有助于完善最终的跟踪结果：第 3 节的跟踪过程对要挑选或丢弃的假设进行了离散选择，这是不可微分的。但是，我们通过对跟踪过程中遇到的所有候选对象（好的和坏的）进行训练模型来帮助它做出最佳选择。换句话说，我们的方法是在训练过程中做出离散选择，然后根据所有可能选择的假设更新参数，这在本质上与使用直通式估计器相似[5]。

虽然原理上很简单，但是必须仔细设计此训练过程以实现最佳性能。我们在此处列出了实施的最重要细节，并研究了它们在消融研究中的影响。

Stopping criterion. 我们以随机的网络权重开始该过程，并在迭代 10 次后训练集大小增加不到 5%时停止该过程。然后，我们在整个训练集上对模型进行全面训练。此过程可以理解为搜索空间的慢速遍历。它从选择随机假设的未经训练的模型开始。然后，随着训练的进行，添加了新的假设，并帮助网络区分好选择和坏选择，并以越来越高的置信度选择最佳选择。

Randomized merging. 在推断过程中，我们通过将每个小轨迹与产生最高分数的轨迹合并来使其生长。相比之下，在训练过程中，我们通过使合并过程随机化来使训练集更加多样化。为此，我们将与合并结果的分数的 softmax 乘以权重系数成正比的概率分配给合并候选者。我们首先设置系数，以便几乎总是选择最佳对，然后逐步减小它以增加随机性。

Balancing the dataset. 一个潜在的困难是，此过程可能导致我们要回归的 IDF 值导致训练集不平衡。通过将数据集按 IDF 值（[0.0; 0.1），[0.1; 0.2），…，[0.9、1.0]），从最小的组中选择所有样本，然后从其他组中选择相同的数字。这使我们能够通过随机选择 h * K 个样本并保留对损耗最大贡献的 K 来执行 h-hard-mining。

5. Results

现在，我们介绍使用的数据集，与之进行比较的基准，我们的结果，最后是定性分析。

5.1. Datasets

我们使用以下公开可用的数据集来对我们的方法进行基准测试。

DukeMTMC. 它包含 8 个序列，包含 50 分钟的训练数据，测试序列为 10 个（困难，密集人群遍历多个摄像机视图）和 25 分钟（简单），具有 60 fps 的隐藏 GT。

MOT17. 它包含 7 个训练测试序列对，它们具有相似的统计信息和测试序列的隐藏 GT，涵盖 785 条轨迹以及静态和动态摄像机。对于每一个，使用不同的算法进行 3 组不同的检测，这使得可以评估跟踪的质量而不会过度适合特定的检测器。

MOT15. 它包含 11 个训练和 11 个测试序列，并具有各种设置的移动和固定摄像机。隐藏了测试的基本事实，并且对于每个测试序列，训练数据中都有一个序列具有大致相似的统计信息。

5.2. Baselines

我们将其与大量 SOTA 算法进行了比较。下面我们将不使用外观提示的与使用外观提示的区分开。

Algorithms that ignore Appearance Cues.

• LP2D 是与 MOT15 一起提供的得分最高的外观少的原始基线。它根据求解线性程序来制定跟踪。

• RNN 依赖于递归神经网络，并且在本质上类似于我们的神经网络，因为它使用 RNN 以直接的方式进行跟踪。 然而，使用不同的损失和方法来训练它以创建训练数据。

•PTRACK 旨在通过优化其他方法产生的轨迹来改善其他方法的结果，以最大程度地近似 IDF 度量。近似值是手动设计的，并非像我们的方法那样学习。

•SORT 将卡尔曼滤波与匈牙利算法结合在一起，目前是 MOT15 数据集上最快的算法。

Algorithms that exploit Appearance Cues.

•MHT 在从卷积姿势机制中提取的姿势特征等辅助下执行多个假设跟踪[51]。

•CDSC 使用控制集聚类来执行摄像机内和摄像机间跟踪。它采用了在 ImageNet 上经过预训练的 ResNet-50 [16]中的图像功能。

•REID 执行 Tracklet 的分层聚类，并利用对 7 个不同数据集进行预训练的[58]的重新识别模型。

•BIPCC 通过解决二进制整数问题对具有相似外观的检测进行聚类。这是 DukeMTMC 数据集的基线方法。

•DMAN 使用双重注意力网络通过关注相关的图像部分和时间片段来执行数据关联。

•JCC 将多个对象跟踪和运动分割作为一个联合共聚问题来处理。它通过局部搜索对像素和边界框进行分组来解决。这将返回轨迹和分段。

•MOTDT17 通过使用学习到的重新识别度量对检测进行分组，利用几何特征和卡尔曼滤波器来执行分层数据关联。

•MHTBLSTM 在精神上类似于我们的方法。它使用多重假设跟踪器和序列模型对跟踪进行评分。但是，仅使用具有最多一个 FP 甚至有时错过检测的 GT 序列对其进行训练。

•EDMT17 依靠多重假设追踪器。 它的成长和修剪阶段取决于学习到的检测检测和检测场景关联模型，这些模型用于更好地对检测和假设进行评分。

•FWT 解决了一个二进制二次问题，可以对分别获得的头部和身体检测进行最佳分组。我们将在第 5 节中展示，当使用与它们相同的设置时，我们的性能优于两种方法。

5.3. Experimental Protocol

在本节中，我们将描述我们在实践中使用的功能以及批处理，训练和选择超参数的方法。

Features. 为了与上述两类基准进行公平比较，我们使用外观不起作用的特征或编码实际图像信息的特征。 我们在下面描述它们。

Appearance-less features. 我们使用以下简单功能，无需进一步参考图像即可从检测结果中进行计算：

•边界框坐标和置信度（$\in R^5$）。

•相对于 tracklet 的上一个和下一个 detection 的边界框偏移（$\in R^8$）。

•Social feature-附近检测结果的描述，$\in R^{3*M}$。 它包括对 M 个最近检测的偏移及其置信度值。所有值均相对于图像大小表示，以实现更好的概括。

Appearance-based features. 作为外观的基础，我们使用[19]的重新识别模型从边界框生成的 128 维向量。这些向量之间在欧式空间中的距离表示人的外观之间的相似性以及他们是同一个人的可能性。为此，我们在基于外观的模型中提供以下附加特征：

•每个边界框的外观向量（$\in R^{128}$）。

•如果有可用的话，则从边界框中的外观到最能代表当前批次之前的轨迹的欧几里得距离（$\in R^1$）。 为了选择到目前为止最能代表轨迹的外观，我们计算了轨迹中每对出现之间的欧几里得距离，并选择了所有其他距离的总和最小的欧几里得距离。

•人群密度特征-从当前边界框的中心到当前帧中最近的第 1 个，第 5 个和第 20 个检测点的中心的距离（$\in R^3$）。正如我们在消融研究中所讨论的那样，该特征在非常密集的人群场景中对模型的行为产生了影响。

Batch processing. 在第 3 节中，我们专注于处理一批 N 幅图像。在实践中，我们通过将较长的序列分割为重叠的批处理，将每个序列移动 N / 3 帧来处理更长的序列。在修剪假设时，我们永远不会抑制所有可以与前一批轨迹合并的假设。这样可以确保我们可以合并上一批中的所有轨迹。我们使用了 3 秒的长批次进行训练，如[48]所示。在推断过程中，我们观察到我们的模型能够泛化超过 3s，并且在长遮挡的情况下具有更长的批次可能是有益的。推理使用了 6 秒长的批处理。

Training and Hyperparameters. 对于所有数据集和序列，交叉验证显示 3.2 节的阈值C_{IOU}和C_{score}等于 0.6 和 4.2 节的困难挖掘参数 h 等于 3 是接近最佳的选择。对于

DukeMTMC，我们为每个摄像机选择了 15'000 帧验证集，同时根据所有摄像机的数据对模型进行了预训练，并对每个序列的训练数据进行了最终训练。我们仅使用 DukeMTMC 训练数据来训练[19]的外观模型。对于每个 MOT15 对训练和测试序列对，我们将训练序列用于验证目的，并使用其余训练序列来学习网络权重。对于 MOT17，我们在 PathTrack 上训练了我们的模型，在 CUHK03 [32]数据集上训练了[19]的外观模型，并使用 MOT17 训练序列进行验证。更多细节在附录中。

5.4. Comparative Performance

我们将 DukeMTMC 和 MOT15 与忽略外观特征的方法进行了比较，因为它们的结果报告在这两个数据集中。出于同样的原因，我们使用 DukeMTMC 和 MOT17 与那些利用外观的产品进行比较。我们总结了以下结果，报告了 IDF 和 MOTA 跟踪指标，以及许多身份交换机（ID），并在附录中提供了更详细的细分。我们在图 3 和图 4 中给出一些跟踪结果。

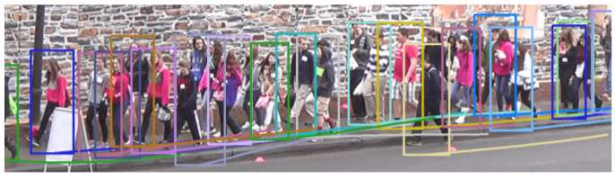


Figure 3. Bounding boxes and the last 6 seconds of tracking, denoted by lines, in dense crowd on **DukeMTMC** dataset.

Figure 4. Bounding boxes and last 6 seconds of tracking, denoted by lines, in two sequences of the **MOT17** dataset.

与利用外观的算法相比。我们在 Tab3 中报告关于 MOT17 的结果，在 Tab1 中报告关于 DukeMTMC 的结果。

Method	App.	**IDF**	**MOTA**	IDs	**IDF**	**MOTA**	IDs
Sequence		Easy			Hard		
OURS	+	**84.0**	**79.2**	**169**	**76.8**	**65.4**	**267**
MHT	+	80.3	78.3	406	63.5	59.6	1468
REID	+	79.2	68.8	449	71.6	60.9	572
CDSC	+	77.0	70.9	693	65.5	59.6	1637
OURS-geom	-	76.5	69.3	426	65.5	59.1	972
PTRACK	-	71.2	59.3	290	65.0	54.4	661
BIPCC	+	70.1	59.4	300	64.5	54.6	652

Table 1. Results on the **DukeMTMC** dataset. The second column indicates whether or not the method uses appearance information.

Method	Use appearance	**IDF**	**MOTA**	IDs
OURS-geom	-	**27.1**	**22.2**	**700**
SORT	-	26.8	21.7	1231
RNN	-	17.1	19.0	1490
LP2D	-	—	19.8	1649

Table 2. Results on the **MOT15** dataset. Appearance is never used.

Method	Use appearance	**IDF**	**MOTA**	IDs
OURS	+	**57.2**	44.2	**1529**
DMAN	+	55.7	48.2	2194
JCC	+	54.5	51.2	1802
MOTDT17	+	52.7	50.9	2474
MHTBLSTM	+	51.9	47.5	2069
EDMT17	+	51.3	50.0	2264
FWT	+	47.6	**51.3**	2648

Table 3. Results on the **MOT17** dataset. Appearance always used.

在 DukeMTMC 上，就 IDF，MOTA 和 IDs 的原始数量而言，我们的方法在 Easy 和

Hard 序列上均表现最佳。此外，与其他使用在其他数据集上预先训练的重新识别网络的最高评分方法不同，我们的方法仅使用 DukeMTMC 训练数据进行训练。

在 MOT17 上，我们的方法在 IDF 度量和 IDs 数量上都是最好的。但是，它在 MOTA 上效果不佳。引人注目的是，FWT 恰好相反：在此数据集上，它产生的 MOTA 最好，IDF 最差。对轨迹的仔细检查表明，这来自产生许多短轨迹，这些轨迹增加了被跟踪检测的总数，因此增加了 MOTA 的数量，但要以分配许多虚假身份，增加碎片化和减少 IDF 为代价。 **这个例子说明了为什么我们认为 IDF 是更有意义的指标，以及为什么我们设计了 tracklet 得分函数来代替它**。

与忽略外观的算法相比。我们在表 2 中报告关于 MOT15 的结果，在表 1 中报告关于 DukeMTMC 的结果。在 MOT15 数据集上，最类似于我们的方法是 RNN，它也使用 RNN 进行数据关联。尽管 RNN 使用外部数据对模型进行预训练，而我们仅使用 MOT15 训练数据，但我们的方法能够以较大的优势胜过它。SORT 的另一个有趣的比较是，它的性能几乎与我们的方法一样好。但是，它不能有效地利用训练数据，并且表明我们还对用于 DukeMTMC 的验证数据运行了该方法，而 DukeMTMC 使用的验证数据要比 MOT15 多得多。这导致 MOTA 得分为 49.9，IDF 为 24.9，而我们的方法在同一数据上分别达到 70.0 和 74.6。

值得注意的是，在 DukeMTMC 数据集上，即使出于比较的目的我们忽略了外观，但我们的方法还是优于或与某些利用它的方法相抗衡[46，49]。这表明我们的训练程序足够强大，可以克服这一严重的障碍。

5.5. Analysis

现在，我们简要分析我们方法的一些关键组成部分，并在附录中提供其他详细信息。

计算复杂度。我们在单个 2.5Hz CPU 上进行了训练，并在 20 个这样的 CPU 上并行执行了所有其他动作（计算 IDF 值以实现数据集平衡，生成训练数据等）。 训练数据最多包含 1.5×10^7 个运动轨迹（DukeMTMC 数据集，摄像机 6），在平衡数据集后最多得到 1.35×10^6 个训练数据点。生成训练数据用时不到 6 个小时，对其进行的训练在 30 个 epoch 内均获得了最佳的验证评分，每个 epoch 要都在 10 分钟以内。 推理速度约为每秒 2 帧。但是，在 3.2.1 节中的修剪步骤中添加序列分数的截止值可将我们的 python 实现速度提高到 30fps，但代价是性能会降低很小（IDF 为 71，而不是 74.6）。

消融研究。 DukeMTMC 的最后 15'000 帧训练序列用于消融研究。我们改变了解决方案的三个主要组成部分，以显示它们对跟踪准确性的影响：**数据组成**，**评分功能**和**训练过程**。应用此类更改时，我们报告 IDF 下降。通过考虑最多具有一个身份切换的小轨迹来创建固定的训练集，如[48、35]中所述，会降低性能（-3.9）。根据分数或总计数来修剪假设[56]会导致计算爆炸或性能下降（-20）。像[48]中那样，根据 S（Φ（T））的预测，直接使 IDF 值回归，不使边界框偏移回归或使用标准分类损失计算损失同样适得其反（-5.1，-13.2，-2.2 ，-32.8）。不平衡训练集或不使用强制挖掘也会对结果产生不利影响（-4.7，-2.5）。使用 Integer 程序而非贪婪算法选择最终解决方案，分别对每种类型的特征进行预训练模型或训练更深的网络都没有明显的效果。

Feature groups。我们还评估了不同特征如何影响解决方案的质量。外观特征将整体 IDF 从 74.6 提高到 82.5，外观距离特征的效果最大。人群密度特征主要影响拥挤的场景，与较少拥挤的场景相比，我们的合并过程更喜欢合并时间间隔较近但在视觉上更相似的检测，相比之下，拥挤的场景则更倾向于基于空间邻近度合并检测。Social feature 主要影响无外观模型，通过确保周围轨迹的检测在整个轨迹中保持一致来帮助保存身份，从而将 IDF 从 67.5 提高到 74.6。4.2 节中的概率合并对于将基于外观的特征和基于几何的特征融合在一起至关重要。没有它，仅选择最佳候选者将导致执行合并的模型主要基

于外观信息（主要忽略空间邻近度）或基于空间和运动信息（主要忽略外观信息）。

#	Δ Dataset	IDF	Δ Loss	IDF	Δ Training	IDF	Δ Tracking	IDF
1	Dataset: all pairs	71.5	Loss on IDF	69.5	-hardmining	72.1	Batch 6	72.6
2	Dataset: mix of two	70.7	Regressing IDF	63.4	-balanced dataset	69.9	IP solution	73.8
3	Selected only	63.6	-bbox regression	72.4	pretraining	71.9		
4	Prunning by score	—	-bbox loss	66.3	2 layer LSTM	74.1		
5	Prunning by count	54.2	classification	41.8				

Table 4. Ablation study. Left, middle and right columns show possible changes in dataset creation procedure, loss function, training and tracking procedure, as well as respective values of **IDF** metric with respect to reference solution (**IDF** 74.6). Details about each change are given in Sec. A.1.

6. Conclusion

我们引入了一种训练过程，通过迭代构建丰富的训练集来显着提高序列模型的性能。我们还开发了一种复杂的模型，该模型可以从小轨迹回归到 IDF 多目标跟踪指标。我们已经表明，在使用外观和不使用外观的情况下，我们的方法在几个具有挑战性的基准上均优于最新方法。在第二种情况下，我们甚至可以接近不使用基于外观的方法所能完成的工作。这对于解决难以使用外观的问题（例如细胞或动物追踪）可能是极其有用的[40]。

在以后的工作中，我们将扩展数据关联过程以考虑更高级的外观功能，例如 2D 和 3D 姿势。我们还将研究通过使用实际的 IDF 代替提议的 IDF 回归器来进一步减少损失评估失配，这将需要使用强化学习。

A. An appendix

A.1. Ablation study

消融研究。它是使用每个摄像机视图中的最后 15000 个训练帧对 DukeMTMC 的验证数据执行的。表 4 将结果分为四列。

•Changes in dataset。 我们通过以下方式量化降低第 3 节的训练数据集生成过程的影响：

1.在所有检测对之间使用随机 tracklets；

2.使用的 tracklets 由至多两个 GT 轨迹组合

3.不是所有的在 growing 阶段观察到的 tracklets 都加到训练数据中，而仅将最终解决方案中存在的 tracklets 添加到训练数据中；

4.使用 tracklet 的预测分数作为截止值进行修剪；

5.通过保持最佳得分的 tracklets 数量固定来进行修剪。

1），2）和 3）产生的训练数据较小且差异较小，这对跟踪结果有不利影响。 4）不允许我们训练任何合理的模型，因为具有非常相似得分的轨迹的计算爆炸，这些都被纳入训练数据中。 5）由于相同的原因被证明无效-训练数据包含许多非常相似的轨迹。