INTEGRATING MOTION PRIORS FOR END-TO-END ATTENTION-BASED MULTI-OBJECT TRACKING

R. Ali*, M. Mehltretter, C. Heipke

Institute of Photogrammetry and GeoInformation, Leibniz University Hannover, Germany
(ali, mehltretter, heipke)@ipi.uni-hannover.de

KEY WORDS: Pedestrian Tracking, Image Sequence Analysis, Attention, Transformer, Motion Modelling.

ABSTRACT:

Recent advancements in multi-object tracking (MOT) have heavily relied on object detection models, with attention-based models like DEtection TRansformer (DETR) demonstrating state-of-the-art capabilities. However, the utilization of attention-based detection models in tracking poses a limitation due to their large parameter count, necessitating substantial training data and powerful hardware for parameter estimation. Ignoring this limitation can lead to a loss of valuable temporal information, resulting in decreased tracking performance and increased identity (ID) switches. To address this challenge, we propose a novel framework that directly incorporates motion priors into the tracking attention layer, enabling an end-to-end solution. Our contributions include: I) a novel approach for integrating motion priors into attention-based multi-object tracking models, and II) a specific realisation of this approach using a Kalman filter with a constant velocity assumption as motion prior. Our method was evaluated on the Multi-Object Tracking dataset MOT17, initial results are reported in the paper. Compared to a baseline model without motion prior, we achieve a reduction in the number of ID switches with the new method.

1. INTRODUCTION

Visual multi-object tracking (MOT) is a crucial task in various real-world applications such as autonomous driving, surveillance, and human-robot interaction. It involves detecting and tracking multiple objects in a video sequence. Many recent advancements in the field of MOT have been dependent on the performance of the employed object detection model. With the advent of transformer models (Vaswani et al., 2017) and transformer-based detection models (Carion et al., 2020), which have demonstrated state-of-the-art capabilities, multiple attention-based tracking models have been developed (Sun et al., 2020, Meinhardt et al., 2022). These tracking models exploit the underlying architecture of transformer-based detection models, particularly the encoder-decoder framework. Those models propagate the so-called detected-queries, which encode information about the position, class and detection score of the detected object from the previous frames as an additional input to the decoder component of the model in the current frame. One of the practical limitations of these tracking models is the limited length of tracks that the model can be trained on due to hardware restrictions, resulting in a loss of temporal information. In a number of approaches, post processing is used to counteract this shortcoming.

To overcome this limitation in a better way, in this paper we propose a novel framework that integrates motion priors directly into the attention layer, facilitating an end-to-end solution. By incorporating motion priors within the model, we eliminate the need for a separate post processing step. We use the motion-prior model used by (Zhang et al., 2022) in which a Kalman filter appro ach (Welch and Bishop, 1995) is used with the help of constant velocity assumption as motion model. Thus, this paper contains the following main contributions:

- A novel approach to integrate motion priors into attention based multi-object tracking models.
- A specific realisation of this approach using a Kalman filter with constant velocity assumption as motion prior.

The structure of the paper is as follows: Chapter 2 discusses the two main tracking paradigms, tracking-by-detection and joint-detection-and-tracking, followed by an overview of the main attention-based detection models. Chapter 3 provides a general explanation of attention-based models, and subsequently focuses on the detection and tracking aspects of our model. Chapter 4 describes the experiments conducted to evaluate our model. Finally, in Chapter 5, we present conclusions and discuss future directions for research.

2. RELATED WORKS

In this section we give a short review of the two main paradigm for MOT: tracking-by-detection and joint-detection-and-tracking, followed by one of the main transformer-based detection approaches which are used for tracking, namely DEtection TRansformer (DETR).

2.1 Tracking-by-Detection

Tracking-by-detection is a widely employed paradigm for multi-object tracking that involves dividing the MOT problem into two distinct steps. Firstly, all objects are detected in each frame separately. In the subsequent so-called re-association step, the detected objects are linked across consecutive frames to establish their trajectories over time. The initial step typically involves employing state-of-the-art object detectors (Ren et al., 2015, Carion et al., 2020, Zhu et al., 2020) to accurately localize all objects of interest. In the re-association step, various methods have been employed to link the detected objects. Trajectory prediction can be achieved through object motion modeling, which may involve employing a simple motion model encoding a constant velocity assumption (CVA), in which the motion is assumed to be of constant velocity over a short period of time.

* Corresponding author

Alternatively a more complex model like a social force model (Pellegrini et al., 2009, Scovanner and Tappen, 2009, Yamaguchi et al., 2011, Nguyen and Heipke, 2020) or a higher-order association of detections to trajectories (Henschel, 2021) or the use of appearance information (Menze et al., 2013) can be used. BYTETrack (Zhang et al., 2022), a state-of-the-art approach, executes the re-association step twice: first, it establishes associations between active tracks and high-scoring detections, followed by associations between tracks that have not yet been assigned a detection in the current frame and low-scoring detections. The motion model is leveraged to predict the positions of the tracks in the current frame. The Intersection over Union (IoU) is then employed to compute a similarity score between the predicted bounding box (BB) and the detected BB. The Hungarian method (Kuhn, 1955) is subsequently utilized to match the tracks with the detected BBs using the calculated similarity scores.

A notable limitation of this re-association approach is the challenge of accurately modeling human motion. Choosing an inappropriate motion model can have a detrimental impact on the quality of the tracking results. Appearance-based re-association methods often use similarity measures given by a siamese neural network (Qian et al., 2017, Yu et al., 2018). Similar to motion modeling-based re-association methods, the Hungarian method is used to match the tracks with the detected BBs. This re-association encounters difficulties in accurately tracking objects in crowded scenarios with numerous object and self-occlusions. Tracking-by-detection methods achieve leading performance, but the separation of the detection and re-association task leads to a model with multiple handcrafted parts that have to be designed for the specific dataset. In our approach, since we use an end-to-end learnable model, we assume that the model can learn the parameters of the used motion model. Thus, a comprehensive fine-tuning of the handcrafted parameters is not needed.

2.2 Joint-Detection-and-Tracking

Joint-detection-and-tracking performs detection and tracking simultaneously within a single stage. In this approach, usually, a detection model is modified such that the information about the detections of the previous frames are propagated to the current frame, such as in (Feichtenhofer et al., 2017, Zhang et al., 2018, Bergmann et al., 2019). In (Zhang et al., 2018) the Faster-R-CNN (Ren et al., 2015) model is modified to achieve tracking. In this context, the underlying detection model has a direct influence on the quality of the tracking results. For this reason and with the advancement of state-of-the-art transformer-based detection models, multiple attention-based tracking models were developed (Sun et al., 2020, Meinhardt et al., 2022). Those models propagate the detections from the previous frames to the current one and use them as additional inputs to the decoder of the detection model.

Due to the end-to-end training nature of Joint-Detection-and-Tracking, the approach requires frames of multiple epochs to achieve object tracking over a sequence. As a result, powerful hardware is necessary to accommodate the computational requirements.

2.3 Object Detection with Transformers

One of the first object detection models that leverages attention mechanisms to detect objects in an image is DEtection TRansformer (DETR) (Carion et al., 2020). DETR extracts input image features with a backbone CNN. These features are then augmented with positional encoding to preserve spatial information before being fed into the Encoder block. The encoder uses self-attention to capture global contextual information. The decoder receives the encoded image features, and in addition the so-called object queries as input. While the number of object queries can vary, all of them have the same fixed dimension d_q. The decoder uses object queries to look for relevant image features and outputs predicted queries, each of which is fed through a feed forward network (FFN) that predicts either a detection (i.e., class-score and bounding box) or ”no object”. An overview of the DETR framework can be seen in Figure 1.

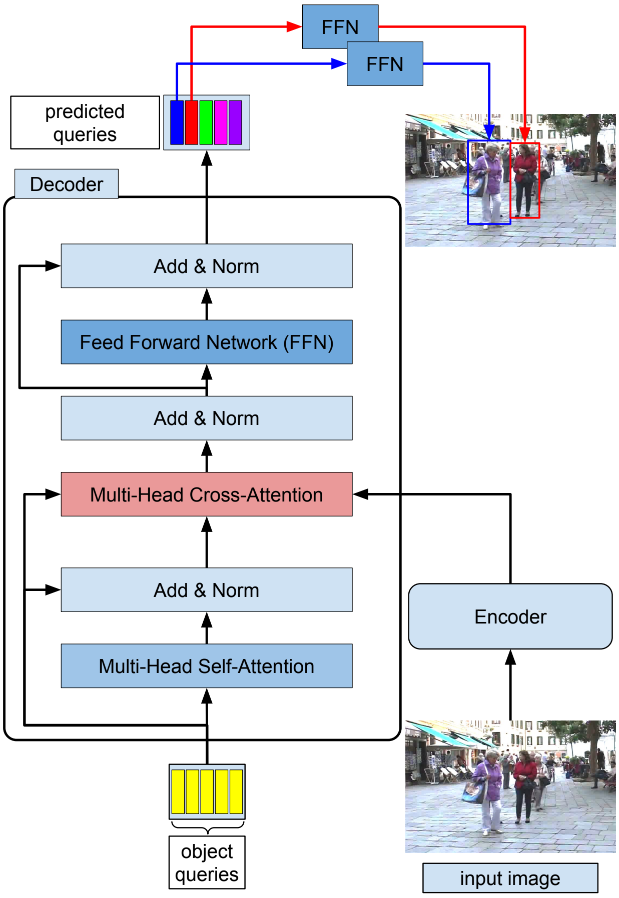

Figure 1. DETR framework that utilizes transformer-based models for object detection. First, a backbone Convolutional Neural Network (CNN) is employed as an encoder to extract image features from the input image. The output of the encoder block, along with the object queries from the previous image, serves as input to the decoder block. The decoder block produces predicted queries, where each query represents a potential object detection. The score associated with each query is compared to a predetermined threshold. If the score surpasses the threshold, it signifies that an object has been identified. In the image, these detected objects are highlighted with colored bounding boxes.

Since the introduction of the DETR model, several approaches have emerged that leverage transformer models for object detection (Zhu et al., 2020, Dai et al., 2021, Gao et al., 2021). Our approach uses the deformable DETR detection model (Zhu et al., 2020) to detect and track objects in a video sequence and falls into the joint-detection-and-tracking category.

3. ATTENTION BASED TRACKING WITH MOTION PRIOR

In this section, we describe the new joint-detection-and-tracking method we have developed. To make the paper more self contained, we start with a review of Transformer models (Vaswani et al., 2017). Then, we explain our tracking method, which is specifically designed for the automatic detection and tracking of pedestrians in an image sequence.

3.1 Review of Transformers

The Transformer model is based on the attention mechanism and was originally developed for natural language processing (NLP, (Vaswani et al., 2017)). More recently it was also employed in computer vision (Dosovitskiy et al., 2021). In essence, transformers model variable and longer range relations between different so-called input tokens (word embeddings in NLP, where an embedding is a linear projection, and embeddings of flattened image tiles in visual transformers). In contrast CNNs have a local and regular neighbourhood, and long range relations can only be established via pooling operations, thus by using additional layers. In visual transformers, relations between all individual pixels of an image can be established by reducing the size of the image tiles. However, a maximum distance is typically introduced to keep the computational complexity under control, e.g. by computing attenion in local windows only (Liu et al., 2021). In particular in datasets with a temporal dimension (sentences in NLP, image sequences), transformers have achieved remarkable results (Brown et al., 2020, Carion et al., 2020, Liu et al., 2021).

It is important to note that the attention mechanism does not involve convolutional layers. This lack of convolutional layers may lead to a loss of spatial information. To address this issue, a positional encoding layer is incorporated, which adds position information to the input tokens.

The attention layer is a crucial component in the Transformer model. The attention function is defined as the mapping of queries zW_q (q for "query") and so-called key-value pairs xW_k and xW_v (k for "key" and v for "value") to produce an output, where $z \in \mathbb{R}^{n_q \times d_q}$ and $x \in \mathbb{R}^{n_k \times d_k}$ are input tokens and $W_v, W_k \in \mathbb{R}^{d_k \times d_k}$ and $W_q \in \mathbb{R}^{d_q \times d_q}$ contain learnable weights with n_q and n_k the number of queries and of values, respectively, and d_q and d_k the dimension of each query and each value, respectively.

Each of these components, queries, keys, and values, is a linear transform of the input tokens z and x. If z and x are identical, the attention function is called self-attention. Conversely, if z and x are differs then the attention function is referred to as cross-attention. The output is computed as a weighted sum of the values, where the weight assigned to each value is determined by a correlation function between its corresponding key and each query, normalised by $\sqrt{d_k}$:

$$Attention(z, x) = softmax(\frac{zW_q \cdot (xW_k)^T}{\sqrt{d_k}}) \cdot xW_v \qquad (1)$$

The attention mechanism can be extended into multiple network heads, here M heads. This extension enables the model to focus on various aspects of attentions between the inputted tokens:

$$MultiHead(z, x) = Concat(H_1, ..., H_M)W_O \qquad (2)$$

where $W_O \in \mathbb{R}^{d_k \times d_k}$ is a learnable linear layer, and each head H_m is calculated as follow:

$$H_m(z, x) = softmax(\frac{zW_{qm} \cdot (xW_{km})^T}{\sqrt{d_k}}) \cdot xW_{vm} \qquad (3)$$

where $W_{vm}, W_{km} \in \mathbb{R}^{d_k \times (d_k/M)}$ and $W_{qm} \in \mathbb{R}^{d_q \times (d_q/M)}$ are learnable weights similar to the one used in single head attention, but with reduced dimension.

Although most transformer models use an encoder/decoder architecture akin to that found in (Vaswani et al., 2017), some models, like (Liu et al., 2021) just use the encoder part. The encoder and decoder components of the transformer model both contain N encoder/decoder blocks, where the encoder blocks only employ self-attention, while the decoder blocks use both self- and cross-attention. Input tokens x are fed to the encoder part of the model, while input tokens z and the encoder output are fed to the decoder. The output of the decoder is a set of tokens that can be interpreted differently based on the task on hand, e.g. in NLP each token can correspond to a word in the dictionary while in object detection in images each token corresponds to a detected object. An overview of the generic transformer model can be seen in Figure 2.

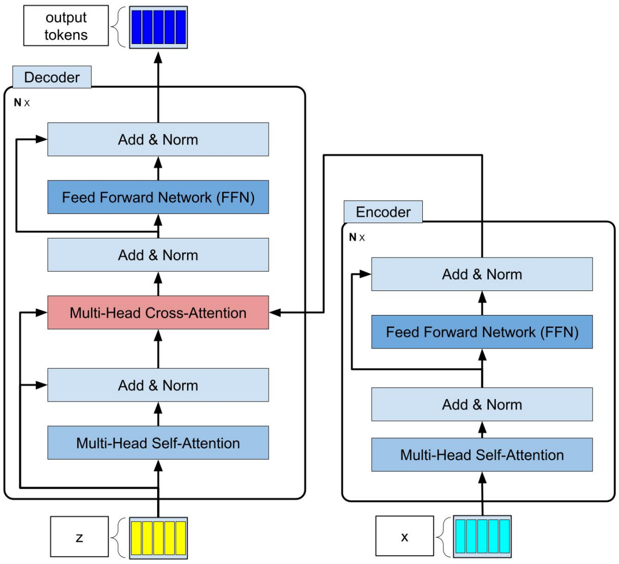
output tokens
Decoder
N x
Add & Norm
Feed Forward Network (FFN)
Add & Norm
Multi-Head Cross-Attention
Add & Norm
Multi-Head Self-Attention
Encoder
N x
Add & Norm
Feed Forward Network (FFN)
Add & Norm
Multi-Head Self-Attention
z
x

Figure 2. Overview of the transformer model where input tokens z and x are used as inputs to the encoder and the decoder, respectively. Both encoder and decoder are made of N encoder/decoder blocks. The output of the encoder is an additional input to the decoder which outputs a set of output tokens.

3.2 Object Detection with Deformable DETR

As mentioend, our tracking method is based on the deformable DETR detection model (Zhu et al., 2020), which is similar to DETR as it is an encoder-decoder model, with N encoder/decoder blocks. Also similar to DETR, a CNN backbone is employed to extract features from the input image; we adopt the feature pyramid network (FPN) (Lin et al., 2017) in our approach to extract image features and thus to be able to represent objects at different scales. Positional encodings are added to these features, before passing them to the encoder block. In the decoder block, the encoded image features and object queries are provided as input, the output of the decoder is a set of predicted queries. Each object query corresponds to a predicted query, which encodes the position, class, and detection score of a potential detection. If the detection score exceeds a predefined threshold, it signifies that the predicted query has successfully identified an object. Conversely, a detection score falling below the threshold indicates the absence of a detection.

The main difference between deformable DETR and DETR is the cross-attention layer. The primary concept behind the cross-attention layer in deformable DETR is related to reducing the computational load by avoiding to compute attentions between every object query z_{nq} and every image feature x (where n denotes the n^{th} decoder block and q is the q^{th} object query), as it is done in DETR. Instead, in deformable DETR attention is computed between each object query z_{nq} and a subset of n_p features selected from the image features x. To accomplish this, three distinct linear layers are utilized to extract a reference point P_q, sampling offsets ΔP_{mqk}, and weights of the attention matrix A_{mqk} from each object query z_n. Here, the reference point P_q is used as the initial guess of the bounding box center, and the sampling offsets ΔP_{mqk} are offsets with respect to P_q. Additionally, k represents the sampling point within the m^{th} attention head and q^{th} object query. This process is applied to all layers of the multi-scale image features obtained through the FPN backbone:

$$\mathrm{MSDeformAttn}\left(z_{nq}, P_q, \left\{x^l\right\}_{l=1}^{L}\right) = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mqkl}\cdot W'_m x^l \left(P_q + \Delta P_{mqkl}\right)\right] \tag{4}$$

where l indexes the input feature level, and L is the total number of input feature levels.

3.3 Object tracking with Deformable DETR

An overview of our model can be seen in Figure 3. In frame $t = 0$, the decoder block takes the encoded image features along with the object queries as input. These object queries are pre-trained fixed-size embeddings that provide information about important image regions. These regions can include areas where pedestrians are present, enabling the model to focus on detecting and tracking these pedestrians. The output of the decoder is a set of predicted queries where each object query corresponds to a predicted query. If the detection score exceeds a predefined threshold, the predicted query has successfully identified an object and this query becomes a track query in the subsequent epoch. In these frames ($t > 0$), the set of object queries is expanded accordingly. Additionally, we use motion priors for the track queries as part of the cross-attention layer of deformable DETR, this will be discussed in subsection 3.4.

The resulting detection queries at this stage can be separated in four types:

- Continued track query: A track is continued if a track query detects its object in the current frame and the detection score is higher than the predefined threshold (see the blue track between frames $t = 1$ and $t = 2$ in Figure 3). In this case, the track query of the previous frame is updated and used as a track query in the next frame.
- Lost track query: A track is lost if a track query from a previous frame detects its object, but the detection score is lower than the predefined threshold. This can happen, for example, if the object gets occluded between frames (see the red track between frames $t = 1$ and $t = 2$ in Figure 3). In this case, the track query at frame $t - 1$ is used as track query in the next frame $t + 1$, without updating it. This is repeated until the track is continued or after a maximum number of repetitions j is reached after which the track is deleted, meaning that the track query is not further propagated to the next frame.
- New track query: A new track query is initialised if an object query detects a new object and its detection score is higher than the predefined threshold (see the orange track at frames $t = 2$ in Figure 3). In this case, this object query is used as new track query in the next frame.
- Background query: An object query that detects an object is labeled as background if the related detection score falls below the predefined threshold.

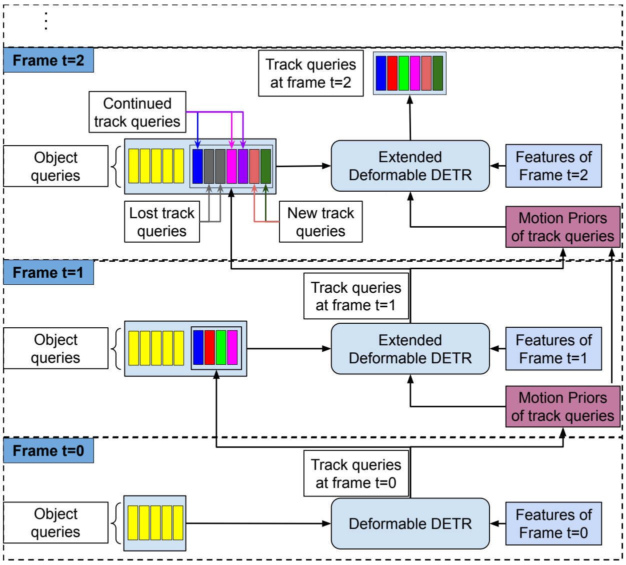

Figure 3. An overview of the proposed method: In frame $t = 0$, the pre-trained object queries are used as input for the tracking model. In subsequent frames ($t > 0$), the detections of frame $t - 1$ are used as input for the tracking model in frame t; these detections are called track queries (for details see text). The motion priors of the track queries are used as additional input to the tracking model.

By using the different types of track queries, namely continued, lost and new track queries, we mitigate the need of using handcrafted re-association models to connect new detections to the existing tracks or to initiate new tracks. Instead, this capability is learned by the tracking model during training. However, a drawback of this approach is that if a specific scenario was not encountered during training, the model may struggle to connect detections to tracks. For instance, if only two frames are used in each training iteration, the model cannot effectively learn about occlusions. This is because learning about occlusions requires at least three frames (preferably more), where the object is detected in the first frame, becomes occluded in the second frame, and reappears in the third frame. To address this challenge, we introduce additional constraints to the model in the form of motion priors.

3.4 Incorporating Motion Priors

Our model integrates motion priors as an additional constraint to enhance the tracking model's ability to predict the position of the tracked object in the next frame based on information from previous frames. This incorporation is achieved within the cross-attention layer, where instead of employing a linear layer to extract the reference point P_q for the track query, in our model we utilize the predicted position of the track query itself as reference point P_q. Additionally, we extend the reference point P_q for each track query to incorporate distinct reference points for different feature levels. This is achieved by incorporating the predicted position of the track query, along with $L-1$ extra points sampled from the normal distributions whose mean and standard deviation are given by the motion prior. Consequently, we obtain P_{lq} reference points, with each reference point corresponding to a specific feature level.

To predict the position of the tracked object in the next frame, we employ a Kalman filter with a constant velocity assumption, following the approach described in (Zhang et al., 2022). The Kalman filtering process consists of two key steps: prediction and update. In the prediction step, the position of the track in each frame is projected forward. The update step is only executed if an object is associated with the track, where the filter is updated based on the available observations.

The Kalman filter with a constant velocity assumption provides estimates of both, the object position and the uncertainty of its position in form of the covariance matrix. The integration of motion priors is achieved within the cross-attention layer of deformable DETR, as illustrated in Figure 4.

If the detected object has moved only slightly between frames, which is usually ensured by high frame rates, a simple motion model like the constant velocity assumption which assumes that an object motion follows a constant velocity over a short period of time, e.g. between the last two frames in which the object has been seen. The speed of each object is recalculated in each frame based on its current and previous position. According to this assumption, the object position changes linearly over time with constant speed and direction; the object's future position can be predicted based on its current position and velocity. The predicted position encourages the model to search for the object at the next frame in the predicted position.

3.5 Loss Function

The loss function used in our approach is identical to the one employed in TrackFormer (Meinhardt et al., 2022). To make this paper self-contained we explain it briefly in the following. The loss is calculated in two steps: First, bipartite matching is employed to establish a mapping $j = \pi(i)$ between the ground truth objects y_i and the combined set of object and track query predictions $\hat{y}_j$. This matching is done similar to the one used in (Carion et al., 2020) where the mapping $j = \pi(i)$ corresponds to the mapping in which $\sum^{R} L_{match}(y_i, \hat{y}_{\pi(i)})$ is minimized, where R is the list of all the indices of the used object and track queries, and $L_{match}(y_i, \hat{y}_{\pi(i)})$ is a pair-wise matching cost between ground truth y_i and a prediction with index $\pi(i)$. Here, L_{match} is an addition to class and bounding box loss.

Second, a set prediction loss is calculated where y_i, $\hat{y}_j$ and the mapping π are used to calculate the loss:

$$\mathcal{L}_{MOT}(y_i, \hat{y}_j, \pi) = \sum_{i=1}^{R} \mathcal{L}_q(y_i, \hat{y}_j, \pi) \tag{5}$$

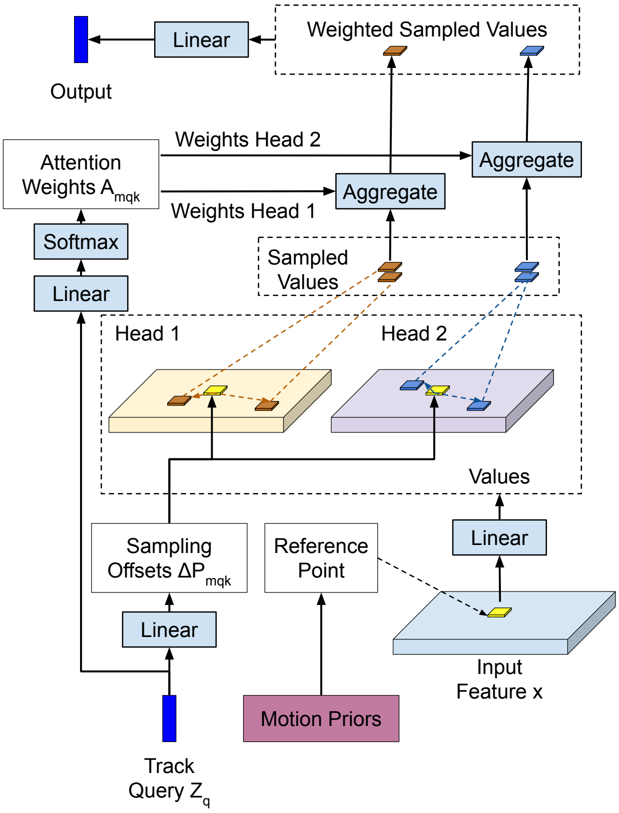

Figure 4. Our modified variant of the multi head cross-attention layer of deformable DETR using motion priors. Initially, values are extracted from the input feature x using a linear layer. The predicted position of the track query serves as the reference point for these values. Subsequently, sampling offsets and attention weights are obtained through two separate linear layers applied to the track query. The values are then sampled by incorporating the reference point, changed by the determined offsets. Finally, the output of the cross-attention layer is derived as the linear transformation of the sum of the weighted sampled values.

If a query is a lost track query or a background query, its loss is calculated as follow: $\mathcal{L}_q = -\log \hat{p}_i(0)$ where just the class, $i.e.$ background, is considered for the loss calculation and not the bounding box. $\hat{p}_i(0)$ denotes the predicted probability of the background class. On the other hand, for all the other queries, namely continued and new track queries, the loss is calculated as follow: $\mathcal{L}_q = -\log \hat{p}_i(c_{\pi=i}) + \mathcal{L}_{\text{box}}(b_{\pi=i}, \hat{b}_i)$ where both class and bounding box losses are calculated. $\mathcal{L}_{\text{box}}(b_{\pi=i}, \hat{b}_i)$ is a combination of the $L1$ distance and the intersection over union (IoU).

4. EXPERIMENTS

In this section, we discuss the experiments conducted to evaluate our model. We compare the achieved results to a baseline that does not utilize motion priors. Instead, the baseline uses the last position of each track query as reference point in the cross-attention layer, identical to the approach described in (Meinhardt et al., 2022).

4.1 Dataset

For training, two datasets where used, the first one *CrowdHuman* (Shao et al., 2018) is a large dataset containing

15k training images with common pose annotations. Stationary cameras are positioned in crowded scenes characterized by various kinds of occlusions. This dataset is used to pre-train a Deformable DETR detection model that we then extend as explained in the Section 3. The second dataset is $MOT17$ (Milan et al., 2016) which is a tracking benchmark containing 14 video sequences. 7 of them are used for training and the other 7 for testing. In this dataset, stationary cameras are positioned in crowded scenes characterized by significant occlusions. The frame rate is between 25 and 30 fps.

4.2 Evaluation Metrics

To quantitatively evaluate the tracking approach, we utilize several metrics commonly employed in the tracking domain: MOTA (Multiple Object Tracking Accuracy) (Bernardin and Stiefelhagen, 2008), IDF1 (Identity-F1 score) (Ristani et al., 2016), and ID switches. These metrics provide a comprehensive assessment of the tracking results. MOTA combines three different error metrics, including identity (ID) switches, false positives, and false negatives, to calculate a single score. By summing up these metrics and dividing the sum by the total number of objects in all frames, we obtain the total error rate E_{tot}. MOTA is then defined as $1 - E_{tot}$. IDF1 specifically evaluates the correctness and consistency of object IDS and trajectories by combining ID precision (IDP) and ID recall (IDR) using the harmonic mean. IDP measures the ratio of true positives to true positives plus false positives, while IDR measures the ratio of true positives to true positives plus false negatives. ID switches measures the number of switches in the track / in the dataset.

4.3 Training Strategy and Implementation Details

The training strategy employed in (Zhu et al., 2020) and (Meinhardt et al., 2022) is adopted here. Since the proposed model uses joint-detection-and-tracking, for each training step at least two frames have to be used, namely frame t and frame $t - m$, where m denotes the difference in frame numbers between the two frames. The model detects the objects in frame $t - m$ and propagates them to frame t. Since we use a motion prior to aid tracking, for each object at least two previous detections are needed to predict its position in frame t. Thus, for each training step, frames $t - 2m$, $t - m$ and t are used. To enrich the scenarios on which the model is trained on, we chose $1 \leq m \leq 10$, where $m \in \mathbb{N}$, which simulates relatively long occlusions. m is randomly sampled for each iteration in training. Additionally, false negative tracks (FN-tracks) are simulated by removing some of the track queries that are used as input to the model at frame t, the removal is done with a probability of p_{FN}. The last scenario that is simulated is the detection of false positive tracks (FP-tracks) in frame $t - m$, which is done by adding FP-queries to the track-queries at frame $t - m$ with a probability of p_{FP}. Those FP-queries are sampled from the detected queries of frame $t - m$ that were classified as background.

In training, a batch size of 2 is used, along with initial learning rates of $2 * 10^{-3}$ for the encoder-decoder and $2 * 10^{-5}$ for the backbone. We employ a model pre-trained on $CrowdHuman$ which is fine-tuned on $MOT17$ for 45 epochs. For fine-tuning, the images were resized to have a maximum height of 600 pixels, keeping the original height to width ratio, due to hardware limitations.

4.4 Results and Discussion

To evaluate the effectiveness of our proposed method, we perform experiments on the pedestrian-tracking dataset $MOT17$. The results can be seen in Table 1. In order to gain a deeper understanding of the impact of motion priors on the tracking model, we present both, detection and tracking results. As depicted in Table 1, in the detection section, there is a slight decrease of 0.3% in recall and 0.4% in precision. This minor decline in the detection performance is reflected in the overall $MOTA$ scores , as $MOTA$ heavily relies on the accurate detection of objects. On the other hand, $IDF1$ and IDS have improved, which shows that the re-association aspect of the model has benefited from the motion prior.

	Detection		Tracking		
Model	Rcll ↑	Prcn ↑	IDF1 ↑	MOTA ↑	IDS ↓
Baseline	71.3%	90.8%	62.2%	63.2%	2832
Our	70.9%	90.5%	63.4%	63.0%	2636

Table 1. Evaluation on $MOT17$ test set. We compare our model with the baseline model in which no motion prior is used. The results are depicting detection and tracking results.

To better understand the impact of motion priors on the tracking model, we present two examples in Figure 5. These examples demonstrate how motion priors aid the tracking model in recovering from object occlusions. In the first example (frames 75 and 83), the baseline tracking model incorrectly assigns the ID of the occluding object to the occluded object, resulting in the occluding object initiating a new track. The second example (frames 282 and 300) shows how IDS of the occluding and occluded objects are switched.

These results demonstrate the improved association capability of the tracking model when incorporating motion priors. However, there is a slight decrease in the detection aspect of the model. This can be observed in Figure 6, where our model fails to detect the tracked object in frame 290, whereas the baseline model successfully detects it. We attribute this effect to our training strategy, specifically the large time gap between detections caused by using frames $t - 2m$, $t - m$, and t during training. This can lead to incorrectly predicted positions and subsequently incorrect reference points in the cross-attention layer.

Furthermore, the integration of the motion covariance in the tracking model presents limitations. This is due to the use of the Kalman filter parameters, which we have taken from (Zhang et al., 2022). As a result, the covariance ellipse tends to remain either very small or relatively large. When the covariance ellipse is small, the sampled points align closely with the predicted position. Conversely, when the covariance ellipse becomes larger, the number of sampled points is often too small to adequately cover the expanded area.

5. CONCLUSION AND FUTURE WORKS

We have proposed a novel tracking approach that incorporates motion priors into an attention-based detection model, adopting the joint-detection-and-tracking paradigm for multi-object tracking. Our proposed framework directly integrates motion priors into the attention layer, enabling end-to-end learning. The experimental results demonstrate an enhancement in the association aspect of the model.

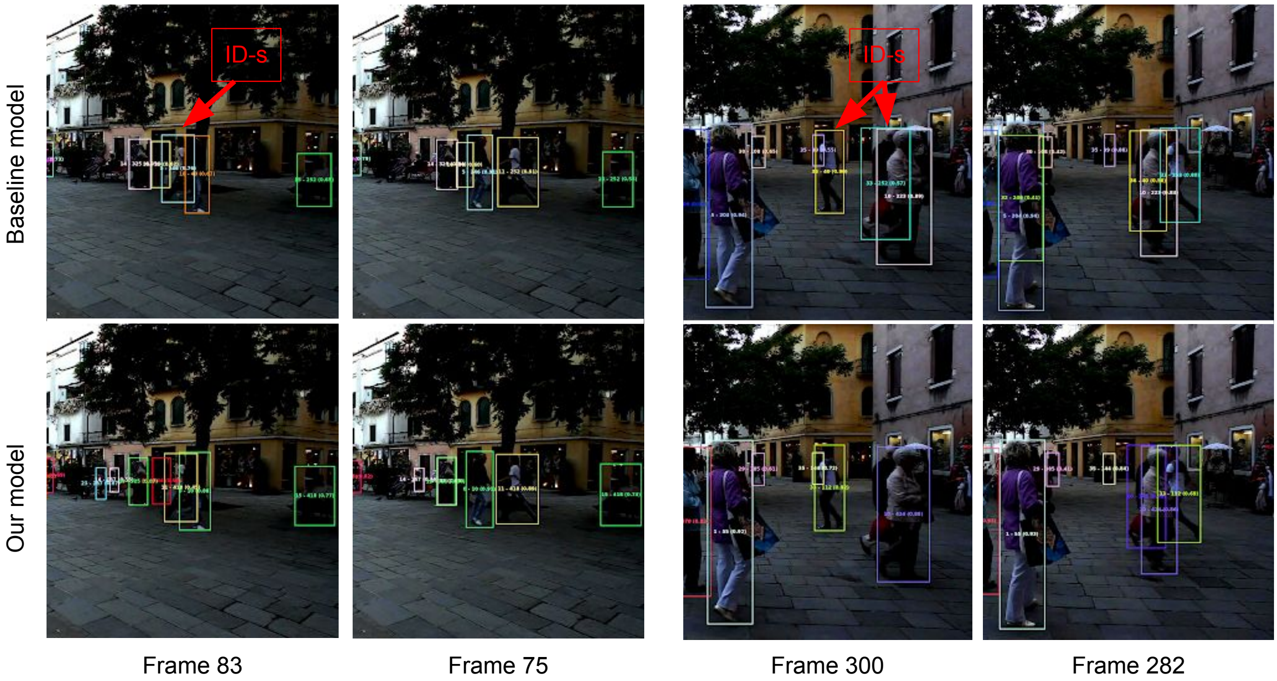

Figure 5. Visualization of results of our model. We show the improvement of our model in comparison with the baseline model. The color of the bounding box denotes the ID of that object. The red arrow denotes an ID-switch.

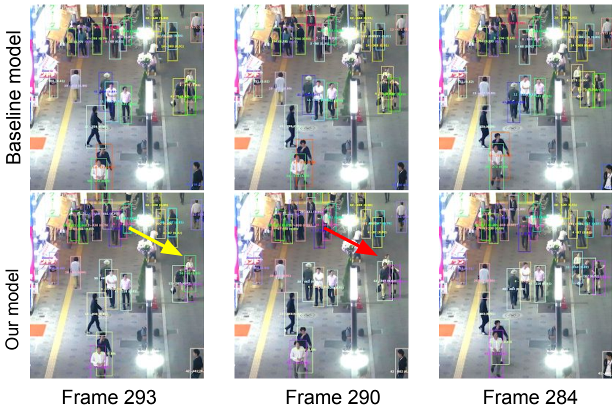

Figure 6. Example of the limitations of our model. The color of the bounding box denotes the ID of that object. The red arrow denotes an FN-track. The yellow arrow denotes the initialisation of a new ID.

In future work we will investigate a two-step training approach. In the first step, the model is trained without the propagation of motion priors, similar to the baseline. This helps to establish a solid foundation for the detection accuracy. In the second step, the motion priors are added. By splitting the training process and limiting the time gap between detections, we then hope to strike a better balance between accurate motion priors and reliable detection performance. Furthermore, the performance of the motion model can be enhanced by improving the parameter selection for the Kalman filter and by exploring alternative motion models, such as learned motion models, to better capture the dynamics of the tracked objects.

6. ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) as a part of the Research Training Group i.c.sens [GRK2159].

REFERENCES

Bergmann, P., Meinhardt, T., Leal-Taixé, L., 2019. Tracking without bells and whistles. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 941–951.

Bernardin, K., Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008, 1–10.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. *Proceedings of the European Conference on Computer Vision (ECCV)*, 213–229.

Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L., 2021. Dynamic detr: End-to-end object detection with dynamic attention. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2988–2997.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*.

Feichtenhofer, C., Pinz, A., Zisserman, A., 2017. Detect to track and track to detect. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3057–3065.

Gao, P., Zheng, M., Wang, X., Dai, J., Li, H., 2021. Fast convergence of detr with spatially modulated co-attention. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3601–3610.

Henschel, R. D., 2021. Higher-order multiple object tracking. PhD thesis, Leibniz University Hannover.

Kuhn, H. W., 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 83-97.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 10012–10022.

Meinhardt, T., Kirillov, A., Leal-Taixé, L., Feichtenhofer, C., 2022. Trackformer: Multi-object tracking with transformers. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8844–8854.

Menze, M., Klinger, T., Muhle, D., Metzler, J., Heipke, C., 2013. A stereoscopic approach for the association of people tracks in video surveillance systems. *Photogrammetrie - Fernerkundung - Geoinformation*, 2013(2), 83-92.

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K., 2016. MOT16: A benchmark for multi-object tracking. http://arxiv.org/abs/1603.00831. arXiv: 1603.00831.

Nguyen, U., Heipke, C., 2020. 3D Pedestrian tracking using local structure constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166, 347-358.

Pellegrini, S., Ess, A., Schindler, K., van Gool, L., 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 261–268.

Qian, X., Fu, Y., Jiang, Y.-G., Xiang, T., Xue, X., 2017. Multi-scale deep learning architectures for person re-identification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 5409-5418.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 91–99.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. *Proceedings of the European Conference on Computer Vision (ECCV)*, 17–35.

Scovanner, P., Tappen, M. F., 2009. Learning pedestrian dynamics from the real world. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 381–388.

Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J., 2018. Crowdhuman: A benchmark for detecting human in a crowd. arXiv:1805.00123.

Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P., 2020. TransTrack: multiple-object tracking with transformer. *arXiv preprint arXiv: 2012.15460*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., Polosukhin, I., 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.

Welch, G., Bishop, G., 1995. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

Yamaguchi, K., Berg, A. C., Ortiz, L. E., Berg, T. L., 2011. Who are you with and where are you going? *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1345–1352.

Yu, Q., Chang, X., Song, Y.-Z., Xiang, T., Hospedales, T. M., 2018. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. *arXiv:1711.08106*.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X., 2022. Bytetrack: Multi-object tracking by associating every detection box. *Proceedings of the European Conference on Computer Vision (ECCV)*, 1–21.

Zhang, Z., Cheng, D., Zhu, X., Lin, S., Dai, J., 2018. Integrated object detection and tracking with tracklet-conditioned detection. *arXiv preprint arXiv:1811.11167*.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2020. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.