Learning Future Object Prediction with a Spatiotemporal Detection Transformer

Adam Tonderski\textsuperscript{1,2}, Joakim Johnander\textsuperscript{1,3}, Christoffer Petersson\textsuperscript{1,4}, and Kalle Åström\textsuperscript{2}

\textsuperscript{1} Zenseact, Sweden
\textsuperscript{2} Centre for Mathematical Sciences, Lund University, Sweden
\textsuperscript{3} Dept. of Electrical Engineering, Linköping University, Sweden
\textsuperscript{4} Dept. of Mathematical Sciences, Chalmers University of Technology, Sweden
adam.tonderski@math.lth.se

Abstract. We explore future object prediction – a challenging problem where all objects visible in a future video frame are to be predicted. We propose to tackle this problem end-to-end by training a detection transformer to directly output future objects. In order to make accurate predictions about the future, it is necessary to capture the dynamics in the scene, both of other objects and of the ego-camera. We extend existing detection transformers in two ways to capture the scene dynamics. First, we experiment with three different mechanisms that enable the model to spatiotemporally process multiple frames. Second, we feed egomotion information to the model via cross-attention. We show that both of these cues substantially improve future object prediction performance. Our final approach learns to capture the dynamics and make predictions on par with an oracle for 100 ms prediction horizons, and outperform baselines for longer prediction horizons.

Keywords: Future object prediction, object detection, transformers

1 Introduction

Autonomous robots, such as self-driving vehicles, need to make predictions about the future in order to plan safely. As a robot executes an action, its surroundings change and this change needs to be estimated in order to gauge the value of the action. Several representations for forecasting have been presented in the literature, such as predicting future RGB-values [31,37,10,22], instance segmentations [30], birds-eye view instance segmentations [19]; and trajectories of seen objects [11]. The latter three tasks involves making predictions about dynamic objects, such as humans, vehicles, or animals. These tasks require the ability to estimate the dynamics of the scene – a challenging problem in itself – and use that for extrapolation [19,30,11]. Furthermore, the tasks are inherently ambiguous and methods need to present multiple hypotheses.

One of the core computer vision problems is the detection of objects in images. It is challenging to pose this problem in a completely learning based manner and methods long tackled the problem indirectly [30,38,23], \textit{i.e.}, resorted
Fig. 1: We explore the problem of future object prediction, where we aim to predict all objects visible in a future frame. We design a novel method that takes a sequence of frames as input ($t-1$ and $t$) and predicts the objects in frame $t+1$. The method is only provided ground truth for the single future frame. Here, we show one predicted object with corresponding attention maps for the input images. Our approach accurately predicts the future pedestrian by attending to him in prior frames, in an emergent form of video tracking.

to surrogate losses and relied on post-processing steps that were not modelled during offline training to make predictions. Carion et al. [3] were the first to propose a learning formulation and neural network architecture that enabled end-to-end learning of this task. Their neural network predicted a set of object hypotheses and a differentiable objective function computed the similarity to the annotations. Several works have since then improved these designs for object detection [52, 33, 9, 26, 12] and adapted it for other computer vision tasks, such as multi-object tracking [32]; 3D object detection [34]; biomedical cell instance segmentation [36]; and visual object tracking [50].

In this work we investigate making predictions about the future. The state of the surroundings in the future is represented in the same way as in object detection, i.e. by a set of confidence, class, and bounding box triplets. In order to tackle this problem, it is natural to divide it into several steps: (i) detecting objects in a sequence of frames; (ii) form tracks and extrapolate trajectories; and (iii) anticipate new objects that might appear. None of these steps are explicit in our approach. Instead, we propose to train a neural network for this problem in an end-to-end manner. To this end, we adopt the formulation proposed by Carion et al. [3]. The neural network architectures proposed in their and in follow-up works make use of a key mechanism: cross-attention. This is the mechanism that enables an instance-level representation, originally referred to as object queries [3]. This mechanism, due to its flexibility, provides a natural way to incorporate two additional sources of information, both crucial for enabling an accurate understanding of the dynamics of the scene. Temporal information, in the form of past images, enables the neural network to better anticipate the motion of dynamic agents. Similarly, the agent’s own motion can significantly affect the contents of future images. We learn how to take this into account by explicitly incorporating information about the ego motion, such as velocity and rotation rate, into the network.

Contributions: Our main contributions are:

5 See Greff et al. [14] for a discourse on different methods of representation.
1. We investigate the problem of future object prediction based on the publicly available nuImages and nuScenes datasets. Our code will be made publicly available to facilitate future work on future object prediction.

2. We propose to directly learn the task with transformers. Our experiments demonstrate the effectiveness of such an approach.

3. We show that attention enables the model to process spatiotemporal information and make better predictions about the future. The model is able to learn this without annotated video, relying only on a single annotated future frame.

4. In a detailed ablation study, we investigate different spatiotemporal mechanisms. We find that sequentially cross-attending to prior frames leads to the best performance. This novel mechanism has the advantage of also having linear complexity in number of frames processed by the network.

5. We find that the sequential cross-attention mechanism can also be used to directly integrate ego-motion information into the network. In our experiments, we show that ego-motion information provides substantial benefits.

6. We demonstrate that the future prediction of a large network can outperform lightweight object detectors that observe the future frame. This indicates trade-off that should be considered in real-time systems.

2 Related Work

**Future Prediction:** Prediction of future video frames in terms of RGB-images has been previously studied with temporal convolution [22], adversarial learning [31], recurrent neural networks [37,40,44], parallel multi-dimensional units [1], variational recurrent neural networks [5], atoms (from system identification) [27], and spatial transformers [28]. This problem is not only interesting in its own right, but enables unsupervised learning of video representations [25]. The works of Villegas et al. [45] and Wu et al. [49] demonstrated that such video prediction may benefit from explicit modelling of dynamic objects.
More closely related to our work is future prediction in terms of more high-level concepts, such as trajectories of dynamic objects. This task has been previously tackled using detect-track-predict paradigm \cite{towards, luc}. Hu et al. \cite{hu} demonstrated the benefits of instead learning this task end-to-end. This was achieved via a differentiable warping \cite{luc} of feature maps to a reference frame with the help of ego motion. The warped feature maps were then processed with spatiotemporal 3D-convolutions. Perhaps most closely related to our work is the work of Luc et al. \cite{luc}. Luc et al. proposed to predict future instance segmentations in an end-to-end fashion. To this end, MaskRCNN \cite{maskrcnn} was modified with a convolutional feature forecasting module. Li et al. \cite{li} noted that the latency in object detectors provide a need of future prediction. The focus of their work was to analyze this aspect and they found that classical tracking-based approaches outperform the end-to-end trained neural network proposed by Luc et al.

**Object Detection with Transformers:** The work of Carion et al. \cite{detr}, DETR, showed how object detection can be learnt end-to-end. They proposed to let a representation of the desired object detection output – a set of object slots – cross-attend a representation of the input – a feature map extracted from the image. A concurrent work \cite{maskrcnn} showed how the cross-attention mechanism could be used to move between different output representations used in prior works, e.g., anchor-boxes of FasterRCNN \cite{faster}, corners of CornerNet \cite{cornernet}, and center points of FCOS \cite{fcos} or CenterNet \cite{centernet}. Since then, several works have improved the effectiveness of transformer-based detectors. Dai et al. \cite{dai} proposed a pre-training stage. Wang et al. \cite{wang} proposed a multi-scale variant. Zhu et al. \cite{zhu} replaced the attention layers with multi-head, multi-feature-level deformable convolution. Dai et al. \cite{dai2} investigated another form of multi-feature-level deformable convolution. Gao et al. \cite{gao} improved the locality of the cross-attention heads via a Gaussian-like weighting. Meng et al. \cite{meng} proposed to change how the transformer makes use of its positional encodings. Amongst the most important effects of these works is the reduction in training time, compared to DETR. The work of Meng et al., for instance, reports a massive reduction: an entire order of magnitude.

**Processing Spatiotemporal Data with Transformers:** A number of works extended the use of transformers into the 3-dimensional video domain. Girdhar et al. \cite{girdhar} lets region-of-interest-pooled feature vectors cross-attend a video feature map, extracted by I3D \cite{agrawal}. The spatiotemporal structure, \textit{i.e.} where in the image and at \textit{what} point in time, is captured in the positional encodings. Jaegle et al. \cite{jaegle} proposed Perceiver, a general neural network architecture able to process data with different structure. Similar to the work of Girdhar et al., the spatiotemporal structure is captured via positional encodings and a cross-attention mechanism lets an output representation gather information from the input data. In their work, Perceiver is used to process 2D feature maps extracted from images, 3D feature maps extracted from videos, and point clouds. Wang et al. \cite{duke} use a similar mechanism for video instance segmentation (VIS). Duke et al. \cite{duke} experiments with spatiotemporal self-attention for video object segmentation (VOS), and propose to use local attention to reduce computational cost in long videos. In contrast to action recognition or point cloud classifica-
In this work, we extend DETR to spatiotemporal data. We experiment with several different mechanisms able to process spatiotemporal data, as illustrated in Figure 3: joint cross-attention \cite{13,21,48}, recurrent cross-attention \cite{32}, and a sequential cross-attention mechanism where previous points in time are processed with different transformer layers. Furthermore, we investigate the addition of ego-motion information to the model. Instead of using ego-motion and geometry to warp feature maps as in the work of Hu et al. \cite{19}, we let a series of cross-attention layers learn how to best make use of this information.

3 Method

We aim to design and train a neural network able to make predictions about the future states of objects in the image plane. First, we formalize this task and introduce a single-frame model – based on DETR \cite{3} – that can be trained directly for this task. Next, we extend this model to be able to process spatiotemporal data, i.e., videos, and capture the dynamics of the scene. Last, we add additional ego-motion information to our proposed neural network. Such information is often readily available from on-board sensors and we hypothesize that this is a powerful cue for the neural network to utilize.

3.1 Learning Future Object Prediction

In standard object detection, we are provided an image at time $t$, $I^t \in \mathbb{R}^{3 \times H_0 \times W_0}$, and tasked with finding the set of objects visible in that image, $O^t = \{O^t_m\}_m$, where $m$ enumerates the objects. Each object $O^t_m$ is described by a vector of class probabilities $c^t_m \in [0, 1]^C$ and a bounding box in the image plane $b^t_m \in \mathbb{R}^4$, so that $O^t_m = \{c^t_m, b^t_m\}$. In this work, we explore the problem of future object prediction. This task is similar to object detection, but instead of detecting objects in the current frame at time $t$, the aim is to predict objects visible at a
Fig. 4: Illustration of the single-frame transformer decoder (top left) and three different spatiotemporal transformers: joint attention (top right), sequential cross-attention (bottom left), and recurrent transformer (bottom left).

future point in time. By design, we remain close to the original object detection task. One of the advantages of that is that we can employ methods for object detection, a well-studied problem. These methods are typically also able to model uncertainties and multiple hypotheses by reporting multiple detections with lower confidence. This uncertainty modelling may be necessary when making predictions about the future.

For future object prediction, we aim to predict a set of objects $O^{t+1}$ visible at time $t + 1$,

$$O^{t+1} = F(\{I^\tau, E^\tau\}_{\tau=1}^t) ,$$

based on images $I^\tau$ and ego-motion information $E^\tau$ at times $\tau = 1, \ldots, t$. As a starting point, we shall consider a method taking only a single image as input. In Section 3.2, the method is extended to take multiple images as input, and in Section 3.3, we extend it to also exploit ego-motion. For a single-frame approach, we formally have

$$O^{t+1} = F(I^t) .$$

Even though a single image does not directly capture any dynamics, the network is free to learn strong priors based on, e.g., the type of object, its rotation, and its surroundings. Consider for instance the pose of a human or whether a vehicle is incoming or drives in the same lane as an ego-vehicle.

In this work, we adopt the transformer-based end-to-end object detection method DETR [3]. In DETR, images are first passed through a convolutional neural network (CNN),

$$x^t = \text{CNN}(I^t) ,$$

providing deep feature maps $x^t \in \mathbb{R}^{D \times H \times W}$. Here, $D$ is the feature dimension and the spatial resolution, $(H, W)$, is typically downsampled relative to the original image size. Next, a transformer encoder further processes the deep feature maps,

$$z^t = \text{Encoder}(x^t) .$$

The transformer has the capacity to add global context and model long-range relationships, providing a rich representation of the image, $z^t \in \mathbb{R}^{D \times H \times W}$. Finally,
a transformer decoder is applied. The decoder has a learnt object slot representation \( y_{\text{init}}^t \in \mathbb{R}^{M \times D} \), corresponding to \( M \) possible objects, that is refined by successively cross-attending the deep image features,

\[
y^t = \text{Decoder}(y_{\text{init}}^t, z^t) .
\]  
(5)

Each slot \( y_m^t \) represents a potential object expected to be visible at time \( t+1 \) and predicted at time \( t \). The object classification scores and bounding box parameters are predicted using two small heads,

\[
\{ c_{m}^{t+1} \}_m = \text{Head}_{\text{class}}(y_m^t) , \quad \{ b_{m}^{t+1} \}_m = \text{Head}_{\text{box}}(y_m^t) .
\]  
(6)

The model structure, corresponding to (3)-(6), is illustrated in Figure 2.

In DETR, the neural network is trained end-to-end for the object detection task by first matching the predicted objects, \( \mathcal{O} \), to annotated objects, \( \mathcal{A} \), and then, based on the matching, compute a loss. For future object prediction, we adopt the same loss, but computed against the annotated objects at time \( t+1 \), \( \mathcal{A}_{t+1} = \{ \mathcal{A}^t_n \}_n \), where \( n \) enumerates the annotated objects. To compute the matching, a matching score is computed between each pair of predicted and annotated objects,

\[
\mathcal{L}^{\text{match}}(A_i^{t+1}, O_j^{t+1}) = \mathcal{L}_{\text{class}}^{\text{match}}(A_i^{t+1}, c_j^{t+1}) + \mathcal{L}_{\text{box}}^{\text{match}}(A_i^{t+1}, b_j^{t+1}) .
\]  
(7)

Essentially, the class and bounding box of each predicted object \( O_j^{t+1} \) is compared to each \( A_i^{t+1} \in \mathcal{A}^{t+1} \). The matching scores are fed into the Hungarian algorithm which then computes the matching. Each predicted object \( O_j^{t+1} \) is assigned an index \( \sigma(j) \), corresponding to a matched annotated object or to background. The objective function is then computed as

\[
\mathcal{L}(A^{t+1}, O^{t+1}) = \sum_j \mathcal{L}_{\text{class}}(A_{\sigma(j)}^{t+1}, c_j^{t+1}) + \mathbb{1}_{\sigma(j) \neq \emptyset} \mathcal{L}_{\text{box}}(A_{\sigma(j)}^{t+1}, b_j^{t+1}) .
\]  
(8)

Here, \( \emptyset \) corresponds to no object. We use the same matching cost functions \( \mathcal{L}_{\text{class}}^{\text{match}} \) and \( \mathcal{L}_{\text{box}}^{\text{match}} \), as well as the objective loss functions \( \mathcal{L}_{\text{class}} \) and \( \mathcal{L}_{\text{box}} \), as in Conditional DETR [33].

### 3.2 Capturing Dynamics with Spatiotemporal Mechanisms

Making predictions about the future requires information about the dynamics. A classical pipeline would typically detect objects in multiple frames and then create tracks from the detections in order to estimate their dynamics. Based on the estimated dynamics, predictions can be made about the future. Given only a single image, it may be possible to make good guesses about the dynamics. For instance, cars typically move along their lane approximately at the speed limit. For an animal, the pose of its body can convey the direction of movement and some information about its speed. However, using a single image to guess the dynamics cannot be done with precision and we can easily find examples
where the guess is completely incorrect. For instance, a vehicle that is moving very slowly or a vehicle that is reversing. We would therefore like to capture spatiotemporal information. To this end, we extend DETR to process sequences of images.

To isolate the effect of the spatiotemporal mechanism, we keep the architecture as close to the single-frame version as possible. When the model is provided with a single image as input, it is identical to the single-frame Conditional-DETR described earlier. The first question to answer is at which stage of the network information from different frames should be merged. There are three natural locations: the backbone, the encoder, and the decoder. We always assume that CNN features are extracted separately from each image and do not consider earlier temporal fusion. This is both due to computational reasons, and because we believe that transformers are better suited for the task of temporal fusion. This still leaves us with a number of options. For instance, we can merge the image features immediately after the CNN encoder, so that $z_t = \text{Encoder}(x^t, x^{t-1})$. Alternatively we can keep the transformer encoder non-temporal, and instead let the transformer decoder directly make use of the features from multiple frames, $y_t = \text{Decoder}(y_{\text{init}}, z^t, z^{t-1})$. Naturally, we do not have to restrict ourselves to a single fusion location, and several of these approaches can be used simultaneously.

**Joint Attention:** The next question is how we should modify the transformer encoder and decoder to make use of the additional inputs. Due to how transformers operate on unordered sets of tokens, one simple approach is to concatenate the different sets of tokens into a single, larger, set of tokens. This means that $z_t = \text{Encoder}(x^t \oplus x^{t-1})$, and $y_t = \text{Decoder}(y_{\text{init}}, z^t \oplus z^{t-1})$. These alternatives are visualized in the top-right part of Figures 3 and 4 respectively. However, pure concatenation removes all information about which tokens belong to which frame. To resolve this issue, we add a sinusoidal temporal positional encoding – analogous with the spatial positional encoding that is already used in DETR. Instead of encoding the absolute pixel position, we encode the temporal offset between frames (in seconds). This type of fusion puts minimal restrictions on the mechanism and should in principle be the most flexible approach. However, this approach hinges on good interactions with the temporal encoding. It is also computationally expensive when applied in the encoder, since the number of token interactions scale quadratically with the number of frames.

**Sequential Cross-Attention:** A potentially more efficient approach is to use sequential cross-attention. Instead of attending to all tokens from all frames at once, we add additional cross-attention layers that attend to the information from previous frames. In the case of the decoder, we simply duplicate the existing cross-attention step for each temporal frame, so that a single decoder layer consists of one self-attention followed by several cross-attentions, see bottom left in Figure 4. We adopt a similar approach for the encoder, and introduce separate cross-attentions after the self-attention, see bottom left in Figure 3. This has some advantages: on one hand it removes the quadratic scaling of computational resources. On the other hand it removes the need for explicit temporal positional
encodings, since the temporal dimension is directly baked into the network architecture. To the best of our knowledge, this is a novel adaptation of the popular attention mechanism. Closest in spirit is perhaps axial attention \cite{17,46}, where each axis (e.g., height, width and time) is processed separately.

**Recurrent Transformer:** We also experiment with a recurrent solution, where the final object embeddings of the previous frame are used as input to the transformer decoder, together with the new features. In principle, this allows the network to learn an explicit tracking solution, where detections in one frame are used as the starting point in the next frame. With this approach, the decoder and encoder can be expressed as $\mathbf{z}^t = \text{Encoder}(\mathbf{x}^t, \mathbf{z}^{t-1})$, and $\mathbf{y}^t = \text{Decoder}(\mathbf{y}_\text{init}, \mathbf{z}^t, \mathbf{y}^{t-1})$. The bottom right parts of Figures 3 and 4 show the recurrent solution, where an additional cross-attention mechanism processes the output of the encoder or decoder, respectively, from the previous time-step.

### 3.3 Incorporating Ego-motion Information

A major challenge in future object prediction is that the motion of dynamic objects is highly correlated with the ego-motion of the camera. Autonomous robots often move and even objects that are stationary in the three-dimensional sense could move drastically in the image plane. It is possible to rely on multiple images to estimate and compensate for the ego-motion using visual odometry (VO) and geometry. In principle, the neural network could learn VO and use it as an internal representation. However, many systems are equipped with additional sensors that capture ego-motion information. We hypothesize that such information could constitute a powerful cue for the network to use.

**Encoding:** While the exact sensor setups and ways of extracting ego motion can vary between different robots, we believe it is reasonable to assume that one has access to at least some subset of position, velocity, acceleration, rotation and rotation rate. Some of this information may be three-dimensional, but in the case of autonomous vehicles it could be projected to the road plane. For maximum flexibility, and generalization to slightly different setups, we make minimal assumptions on which ego-motion information is available. Our only processing is to convert information into a local coordinate system. For example global position or rotation is converted into a transformation relative to the last frame. We concatenate all available information and pass it through a 2-layer MLP to create an encoded ego-motion vector

$$\mathbf{f}_\text{ego} = \text{MLP}(\text{translation} \oplus \text{speed} \oplus \text{rotation} \oplus \ldots),$$

where $\oplus$ denotes vector concatenation.

**Fusion Mechanism:** The question of where to incorporate the ego-motion information into the network is largely similar to the previously studied question of where to include information from past images. Since we strive to put minimal assumptions on what information is available and how it should be used, we simply use sequential cross-attention to incorporate the encoded ego-motion vector. This naturally allows for joint fusion with past images, where
$y^t = \text{Decoder}(y^\text{init}, z^t, z^{t-1}, f^t_{\text{ego}}, f^{t-1}_{\text{ego}})$ or $z^t = \text{Encoder}(x^t, x^{t-1}, f^t_{\text{ego}}, f^{t-1}_{\text{ego}})$. We experiment with a third approach where the ego-motion information is added to the features, much like the positional encodings used in attention. That is, $z^t = \text{Encoder}(x^t + f^t_{\text{ego}}, x^{t-1} + f^{t-1}_{\text{ego}})$.

4 Experiments

We implement the different approaches in PyTorch [35] and experiment on the NuScenes and NuImages datasets [2]. The code will be made publicly available to facilitate future research on future object prediction or for other tasks with the models used in this work. First, we introduce the DETR-based approach, trained for future object prediction, together with different spatiotemporal extensions. Second, we introduce ego-motion-information to the model. Third, we compare the performance to a naïve approach, a tracking-based approach, the mechanism proposed by Luc et al. [30], and an oracle. We also provide a comparison for different levels of future prediction. Last, we compare the performance to a lightweight oracle.

4.1 Experimental Setup

NuImages and NuScenes: Although we need video datasets, there is no need for the entire video to be annotated. Both training and evaluation is possible with a single frame of the video being annotated. To this end, we experiment with NuScenes [2] and the subsequent dataset NuImages. NuImages contains samples of 13 images collected at 2 Hz; ego-motion information for each image; and object detection annotations for the 7th image. To simplify the problem we only consider front camera images, leaving us with 13066 training samples and 3215 validation samples. NuScenes is primarily intended for 3D object detection using the full sensor suite, and provides 3D annotations at 2Hz. However, projected 2D bounding boxes are also available. NuScenes is inconsistent in the camera sampling frequency, since the 20Hz camera is resampled to 12Hz. During training we can be flexible with the sampling frequency, leaving us with a training set of 26k front camera samples. However, during evaluation, we discard samples which do not match the desired sampling frequency exactly. This leaves 5630 samples at 2Hz, 5759 samples at 4Hz, 2913 samples at 10Hz and 2884 samples at 20Hz. NuImages and NuScenes has two major advantages for future object prediction – advantages that are natural for datasets aimed towards autonomous robots. First, each dataset is captured with a single sensor suite, avoiding for instance issues with different cameras. Second, ego-motion information is included, which we hypothesise is a powerful cue for future object prediction.

Implementation Details: We adopt ConditionalDETR [33] for our experiments due to its fast training. We use a ResNet50 [16] backbone, pre-trained on ImageNet [10], and the same encoder and decoder hyperparameters as in ConditionalDETR. We train the approaches for future object prediction using the same set-prediction loss function [33]. We optimize with AdamW [29], using a
Table 1: Future object prediction performance for different spatiotemporal mechanisms on the NuImages validation set. The best mechanism is marked in red and the second best in blue.

| Method                | AP50 Mean | AP50 Car | AP50 Pedestrian | AP50 Small Medium Large Mean | AP Car | AP Pedestrian |
|-----------------------|-----------|---------|----------------|-----------------------------|--------|--------------|
| Singleframe           | 13.5      | 22.5    | 6.2            | 3.0                        | 12.9   | 30.8         | 3.8 | 6.7 | 1.5        |
| Joint Attention Encoder| 22.9      | 34.8    | 10.5           | 3.8                        | 23.1   | 49.2         | 7.4 | 11.8 | 2.7        |
| Joint Attention Decoder| 23.5    | 35.4    | 11.0           | 4.2                        | 23.7   | 52.0         | 7.9 | 11.9 | 2.8        |
| Sequential CA Encoder | 21.9      | 35.0    | 9.7            | 4.7                        | 21.9   | 48.1         | 7.1 | 11.9 | 2.5        |
| Sequential CA Decoder | 23.9      | 36.6    | 12.0           | 4.1                        | 25.1   | 52.1         | 8.1 | 12.1 | 3.1        |
| Recurrent Tr. Encoder | 21.6      | 34.9    | 9.5            | 3.7                        | 21.9   | 44.2         | 7.3 | 11.8 | 2.4        |
| Recurrent Tr. Decoder | 21.2      | 32.6    | 9.2            | 3.6                        | 22.1   | 42.4         | 6.6 | 10.8 | 2.3        |

We use base learning rates of $10^{-5}$ for the backbone and $10^{-4}$ for the encoder, decoder, and linear heads. The learning rates are warmed up for the first 10% epochs and decayed by total factors of $\{0.5, 0.1\}$ when 60% and 90% of epochs have been processed. During the first 60% epochs, we train with half resolution and double batch size. We train for 400 epochs on NuImages and 160 epochs on the relatively larger NuScenes.

**Evaluation:** We aim to predict objects present in a future frame. Thus, the detectors are fed images and expected to produce detections for one time-step into the future. Performance is measured in terms of Average Precision (AP). We report both average precision with a 0.5 IoU threshold (AP50) and averaged over the thresholds $\{0.50, 0.55, \ldots, 0.95\}$ (AP). We report both an average score over all classes and for the car and pedestrian classes. We also report results for objects of different sizes, using the cutoffs $H_0/24 \cdot W_0/64$ and $H_0/4 \cdot W_0/12$, where $(H_0, W_0)$ is the original image resolution.

4.2 Quantitative Results

**Spatiotemporal Ablation:** In Section 3, we introduced three mechanisms that each enables the neural network to process video data. Each mechanism can be applied either in the encoder or the decoder. We compare the different alternatives and report the results in Table 1. Compared to the single-frame version, trained for future prediction but with just a single frame, all spatiotemporal variants improve performance substantially. This indicates that each of the three different mechanisms learns to capture and make use of the dynamics in the scene. The best results are achieved by sequential cross-attention placed in the decoder, obtaining 23.9 class-averaged AP50. The joint attention mechanism placed in the decoder achieves competitive performance, 23.5 class-averaged AP50. The recurrent transformer provides inferior performance. Interestingly, joint and sequential attention work best in the decoder, while the recurrent mechanism obtain the best results in the encoder. Based on these results, we adopt the sequential cross-attention mechanism and place it in the decoder.

**Ego-motion Ablation:** Next, we introduce ego-motion information to our spatiotemporal neural network. In Section 3.3, we described how to encode the...
Table 2: Future object prediction performance on the NuImages validation set with and without ego-motion.

| Method                              | AP50 Mean | Car Pedestrian Small | Medium | Large | AP Mean | Car Pedestrian |
|-------------------------------------|-----------|----------------------|--------|-------|---------|----------------|
| No ego-motion                       | 23.9      | 36.6                 | 12.0   | 4.1   | 25.1    | 52.1           | 8.1            | 12.1           | 3.1            |
| Add ego-motion to features          | 28.9      | 42.3                 | 14.7   | 6.8   | 30.5    | 55.6           | 10.1           | 14.9           | 4.0            |
| Attend ego-motion in encoder        | 28.0      | 43.2                 | 15.1   | 4.7   | 29.9    | 59.3           | 9.4            | 15.5           | 3.4            |
| Attend ego-motion in decoder        | 28.3      | 42.4                 | 13.7   | 4.7   | 29.8    | 59.1           | 10.0           | 14.9           | 3.6            |

ego-motion information and incorporate it into the network using sequential cross-attention. As an alternative, we also interpret $f_{\text{ego}}$ as a learnable positional encoding, and simply add it to the CNN features, as is typically done with positional encodings [3]. We evaluate the effectiveness of these approaches in Table 2.

We can clearly see that ego-motion is a powerful cue, and all approaches strongly outperform the method without ego-motion. Interestingly, even the simple mechanism of interpreting it as a positional encoding works very well. We choose to prioritize performance on the Car and Pedestrian categories, and therefore adopt sequential cross-attention in the encoder in further experiments.

Quantitative Comparison: We compare our single-frame and spatiotemporal approach to a naïve baseline, a tracking-based baseline, and an adaption of prior art. The naïve baseline corresponds to the scenario where no future prediction is done. We run standard object detection on the past image ($I_t$) and treat the detections as the future prediction. The tracking-based baseline follows the baseline used in FIERY [19]: The centerpoint distance between pairs of detections in two different frames is used as cost and then the Hungarian algorithm is used to minimize the total cost. Matched objects are extrapolated linearly into the future frame. While it was not possible to directly compare against prior work, we found F2F [30] to be the most applicable. We keep our DETR-like approach but replace the transformer encoder with the F2F module, which takes features from multiple frames and learns to forecast a future feature map. Finally, the oracle acts as a best-case reference, providing the performance of a single-frame detector applied directly to the future frame.

The results are shown in Table 3. The naïve approach fares very poorly, indicating large changes in the scene. The tracking solution improves the performance across the board, but still struggles. Likely, the simple matching and extrapolation heuristics prove insufficient over the large prediction horizon. The F2F adaptation proves much stronger, in several cases more than twice as good as the tracking solution. Interestingly, despite making use of multiple frames, it is consistently beaten by our Singleframe + Ego solution. This clearly demonstrates the advantage of incorporating ego-motion. Finally, our spatiotemporal version outperforms the other approaches for all but the smallest objects. However, we note that there is still a substantial gap to the oracle, likely due to the stochastic nature of the future.

Part of the motivation behind future prediction is to compensate for the latency induced by the object detector. For the relatively long horizon of 500 ms,
Table 3: Performance for future object prediction on the nuImages validation set in terms of AP50 and AP (higher is better). The prediction horizon is 500 ms.

| Method               | AP50 Mean | Car | Pedestrian | Small | Medium | Large | Mean | Car | Pedestrian |
|----------------------|-----------|-----|------------|-------|--------|-------|------|-----|------------|
| Oracle               | 67.5      | 87.2| 73.5       | 42.0  | 69.5   | 84.7  | 38.4 | 56.2| 35.7       |
| Naïve                | 9.8       | 14.5| 2.5        | 3.6   | 9.4    | 18.8  | 3.0  | 4.2 | 0.6        |
| Tracking             | 14.8      | 16.9| 3.6        | 3.3   | 14.8   | 25.9  | 4.5  | 5.0 | 0.8        |
| F2F adaptation [30] | 21.6      | 35.6| 10.8       | 3.3   | 22.0   | 42.0  | 7.2  | 12.2| 2.7        |
| Singleframe (Ours)   | 13.5      | 22.5| 6.2        | 3.0   | 12.9   | 30.8  | 3.8  | 6.7 | 1.5        |
| Spatiotemporal (Ours)| 23.9      | 36.6| 12.0       | 4.1   | 25.1   | 52.1  | 8.1  | 12.1| 3.1        |
| Singleframe + Ego (Ours)| 25.6 | 38.1| 12.8       | 5.1   | 26.9   | 54.8  | 8.9  | 13.2| 3.4        |
| Spatiotemporal + Ego (Ours)| 28.0| 43.2| 14.7       | 4.7   | 29.9   | 59.3  | 9.4  | 15.5| 3.8        |

Table 4: Performance for future object prediction on the NuScenes validation set in terms of AP50 (higher is better) for multiple prediction horizons.

| Method                  | 50 ms Car Pedestrian | 100 ms Car Pedestrian | 250 ms Car Pedestrian | 500 ms Car Pedestrian |
|-------------------------|----------------------|-----------------------|-----------------------|-----------------------|
| Oracle                  | 71.4                 | 49.9                  | 69.7                  | 46.5                  |
| Naïve                   | 70.4                 | 41.7                  | 64.6                  | 25.6                  |
| Tracking                | 70.9                 | 43.6                  | 66.1                  | 36.6                  |
| Singleframe + Ego (Ours)| 71.3                 | 47.7                  | 69.6                  | 44.1                  |
| Spatiotemporal + Ego (Ours) | 72.4 | 48.3| 70.1       | 44.2                  | 65.6                  | 35.8                  | 54.9                  | 21.5                  |

compared to typical network latencies, we have seen that the zero-latency oracle has much better performance. Table 4 compares the performance of our future predictions with the oracle, for a number of prediction horizons. As expected, the gap shrinks for shorter horizons. More surprisingly, we find a threshold, near 100ms, where the future predictions are roughly on par with the oracle. In the case of cars, we even outperform the oracle, likely thanks to the addition of ego-motion and spatiotemporal cues. These results suggest an additional element to the traditional latency-accuracy tradeoff – the future prediction horizon.

4.3 Qualitative Results

In Figure 5, we visualize the attention maps for two sequences. In each sequence, we show a single predicted, future object. For the predicted object, we visualized the attention maps from the object slot to different regions in the two input images. For another example, see Figure 1. In all scenarios, our approach manages to find and attend to the same object in multiple frames. Using only a single annotated frame during training, a rudimentary form of tracking seems to emerge. In Figure 6, we show qualitative results of our spatiotemporal approach on three challenging sequences. These three sequences exhibit ego-motion, moving objects, crowded scenes, and adverse weather condition. Even in such challenging conditions, the approach manages to accurately predict future object states.
5 Conclusion

We have explored the task of future object prediction. A transformer-based object detector was trained for this task and adapted to process videos in order to capture the dynamics in the scene. The cross-attention mechanism provides a straightforward means to process videos and several different mechanisms were investigated. All provided substantial performance improvements over the baseline. Using sequential cross-attention lead to the best performance. Furthermore, the addition of ego-motion information was investigated. Ego-motion was successfully fed to the neural network either in a fashion similar to positional encodings, or via additional cross-attention mechanisms. This yielded major performance improvements for both the single-frame model and for the spatiotemporal model. We hope for this work to serve as a basis for future work on future object prediction.
Acknowledgements: This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC), partially funded by the Swedish Research Council through grant agreement no. 2018-05973.
References

1. Byeon, W., Wang, Q., Srivastava, R.K., Koumoutsakos, P.: Contextvp: Fully context-aware video prediction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 753–769 (2018)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: muscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
5. Castrejon, L., Ballas, N., Courville, A.: Improved conditional vrnns for video prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7608–7617 (2019)
6. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In: Conference on Robot Learning, pp. 86–99. PMLR (2020)
7. Chi, C., Wei, F., Hu, H.: Relationnet++: Bridging visual representations for object detection via transformer decoder. Advances in Neural Information Processing Systems 33, 13564–13574 (2020)
8. Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L.: Dynamic detr: End-to-end object detection with dynamic attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2988–2997 (2021)
9. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1601–1610 (2021)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: Sstvos: Sparse spatiotemporal transformers for video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5912–5921 (2021)
12. Gao, P., Zheng, M., Wang, X., Dai, J., Li, H.: Fast convergence of detr with spatially modulated co-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3621–3630 (2021)
13. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 244–253 (2019)
14. Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Ho, J., Kalchbrenner, N., Weissenaeb, D., Salimans, T.: Axial attention in multi-dimensional transformers (2019)
18. Hong, J., Sapp, B., Philbin, J.: Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8454–8462 (2019)
19. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: Fiery: Future instance prediction in bird’s-eye view from surround monocular cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15273–15282 (2021)
20. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems 28 (2015)
21. Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: International Conference on Machine Learning. pp. 4651–4664. PMLR (2021)
22. Kalchbrenner, N., Oord, A., Simonyan, K.,Danihelka, I., Vinyals, O., Graves, A., Kavukcuoglu, K.: Video pixel networks. In: International Conference on Machine Learning. pp. 1771–1779. PMLR (2017)
23. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV). pp. 734–750 (2018)
24. Li, M., Wang, Y.X., Ramanan, D.: Towards streaming perception (2020)
25. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction. In: proceedings of the IEEE international conference on computer vision. pp. 1744–1752 (2017)
26. Liu, F., Wei, H., Zhao, W., Li, G., Peng, J., Li, Z.: Wb-detr: Transformer-based detector without backbone. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2979–2987 (2021)
27. Liu, W., Sharma, A., Camps, O., Sznaier, M.: Dyan: A dynamical atoms-based network for video prediction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 170–185 (2018)
28. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4463–4471 (2017)
29. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam (2018)
30. Luc, P., Couprie, C., Lecun, Y., Verbeek, J.: Predicting future instance segmentation by forecasting convolutional features. In: Proceedings of the european conference on computer vision (ECCV). pp. 584–599 (2018)
31. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016), http://arxiv.org/abs/1511.05440
32. Meinhardt, T., Kirillov, A., Leal-Taiixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. arXiv preprint arXiv:2101.02702 (2021)
33. Meng, D., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-
performance deep learning library. Advances in neural information processing systems 32 (2019)

36. Prangemeier, T., Reich, C., Koeppl, H.: Attention-based transformers for instance segmentation of cells in microstructures. In: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 700–707. IEEE (2020)

37. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604 (2014)

38. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)

39. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28 (2015)

40. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International conference on machine learning. pp. 843–852. PMLR (2015)

41. Styles, O., Sanchez, V., Guha, T.: Multiple object forecasting: Predicting future object locations in diverse environments. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 690–699 (2020)

42. Tang, C., Salakhutdinov, R.R.: Multiple futures prediction. Advances in Neural Information Processing Systems 32 (2019)

43. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)

44. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. In: 5th International Conference on Learning Representations, ICLR 2017. International Conference on Learning Representations, ICLR (2017)

45. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: international conference on machine learning. pp. 3560–3569. PMLR (2017)

46. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation (2020)

47. Wang, T., Yuan, L., Chen, Y., Feng, J., Yan, S.: Pnp-detr: towards efficient visual analysis with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4661–4670 (2021)

48. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8741–8750 (2021)

49. Wu, Y., Gao, R., Park, J., Chen, Q.: Future video synthesis with object motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5539–5548 (2020)

50. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10448–10457 (2021)

51. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)

52. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2020)