Translating Images into Maps

Avishkar Saha[1], Oscar Mendez[1], Chris Russell[2], Richard Bowden[1]

Abstract— **We approach instantaneous mapping, converting images to a top-down view of the world, as a translation problem. We show how a novel form of transformer network can be used to map from images and video directly to an overhead map or bird's-eye-view (BEV) of the world, in a single end-to-end network. We assume a 1-1 correspondence between a vertical scanline in the image, and rays passing through the camera location in an overhead map. This lets us formulate map generation from an image as a set of sequence-to-sequence translations. Posing the problem as translation allows the network to use the context of the image when interpreting the role of each pixel. This constrained formulation, based upon a strong physical grounding of the problem, leads to a restricted transformer network that is convolutional in the horizontal direction only. The structure allows us to make efficient use of data when training, and obtains state-of-the-art results for instantaneous mapping of three large-scale datasets, including a 15% and 30% relative gain against existing best performing methods on the nuScenes and Argoverse datasets, respectively.**

I. Introduction

Many tasks in autonomous driving are substantially easier from a top-down, map or bird's-eye view (BEV). As many autonomous agents are restricted to the ground-plane, an overhead map is a convenient low-dimensional representation, ideal for navigation, that captures relevant obstacles and hazards. For scenarios such as autonomous driving, semantically segmented BEV maps must be generated on the fly as an instantaneous estimate, to cope with freely moving objects and scenes that are visited only once.

Inferring BEV maps from images requires determining the correspondence between image elements and their location in the world. Multiple works guide their transformation with dense depth and image segmentation maps [1]–[5], while others [6]–[10] have developed approaches which resolve depth and semantics implicitly. Although some exploit the camera's geometric priors [8]–[10], they do not explicitly learn the interaction between image elements and the BEV-plane.

Unlike previous approaches, we treat the transformation to BEV as an image-to-world translation problem, where the objective is to learn an alignment between vertical scan lines in the image and polar rays in BEV. The projective geometry therefore becomes implicit to the network. For our alignment model, we adopt transformers [11], an attention-based architecture for sequence prediction. With its attention mechanisms, we explicitly model pairwise interactions between vertical scanlines in the image and their polar BEV projections. Transformers are well-suited to the image-to-BEV transformation problem, as they can reason about interdependence between objects, depths and the lighting of the scene to achieve a globally consistent representation.

We embed our transformer-based alignment model within an end-to-end learning formulation which takes as input a monocular image with its intrinsic matrix, and predicts semantic BEV maps for static and dynamic classes.

The contributions of our paper are (1) We formulate generating a BEV map from an image as a set of 1D sequence-to-sequence translations. (2) By physically grounding our formulation we construct a restricted data-efficient transformer network that is convolutional with respect to the horizontal x-axis, yet spatially-aware. (3) By combining our formulation with *monotonic attention* from the language domain, we show that knowledge of what is below a point in an image is more important than knowledge of what is above it for accurate mapping; although using both leads to best performance. (4) We show how axial attention improves performance by providing temporal awareness and demonstrate state-of-the-art results across three large-scale datasets.

II. Related Work

BEV object detection: Early approaches detected objects in the image and then regressed 3D pose parameters [12]–[17]. The Mono3D [18] model instead generated 3D bounding box proposals on the ground plane and scored each one by projecting into the image. However, all these works lacked global scene reasoning in 3D as each proposal was generated independently. OFTNet [19] overcame this by generating 3D features from projecting a 3D voxel grid into the image, and performing 3D object detection over those features. While it reasons directly in BEV, the context available to each voxel depends upon its distance from the camera, in contrast, we decouple this relationship to allow each BEV position access to the entire vertical axis of the image.

Inferring semantic BEV maps: BEV object detection has been extended to building semantic maps from images for both static and dynamic objects. Early work in road layout estimation [1] performed semantic segmentation in the image-plane and assumed a flat world mapping to the ground plane via a homography. However, as the flat world assumption leads to artifacts for dynamic objects such as cars and pedestrians, others [3]–[5] exploit depth and semantic segmentation maps to lift objects into BEV. While such intermediate representations provide strong priors, they require image depth and segmentation maps as additional input.

Several works instead reason about semantics and depth implicitly. Some use camera geometry to transform the image into BEV [8]–[10] while others learn this transformation implicitly [2], [6], [7]. Current state-of-the-art approaches

[1] Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, UK, {a.saha, o.mendez, r.bowden}@surrey.ac.uk

[2] *Amazon*, Tubingen, Germany. cmruss@amazon.com

can be categorised as taking a 'compression' [8], [10] or 'lift' approach [9], [20] to the transformation. 'Compression' approaches vertically condense image features into a bottleneck representation and then expand out into BEV, thus creating an implicit relationship between an object's depth and the context available to it. This increases its susceptibility to ignore small, distant objects. 'Lift' approaches instead expand each image into a frustum of features to learn a depth distribution for each pixel. However, each pixel is given the entire image as context, potentially increasing overfitting due to redundancies in the image. Furthermore, neither approaches have spatial awareness, meaning they are unable to leverage the structured environments of urban scenes. We overcome issues with both these approaches by (1) maintaining the spatial structure of the image to explicitly model its alignment with the BEV-plane and (2) adding spatial awareness which allows the network to assign image context across the ray space based on both content and position.

Encoder-decoder transformers: Attention mechanisms were first proposed by Bahdanau *et al.* [21] for machine translation to learn an alignment between source and target sequences using recurrent neural networks (RNNs). Transformers, introduced by Vaswani *et al.* [11], instead implemented attention within an entirely feed-forward network, leading to state-of-the-art performance in many tasks [22], [23].

Like us, the 2D detector DETR [24] performs decoding in a spatial domain through attention. However, their predicted output sequences are sets of object detections, which have no intrinsic order to them, and permits the use of attention's permutation invariant nature without any spatial awareness. In contrast, the order of our predicted BEV ray sequences is inherently spatial and so we need spatial awareness and therefore permutation equivariance in our decoding.

III. METHOD

Our goal is to learn a model $\mathbf{\Phi}$ that takes a monocular image $\mathbf{I}$ and produces a semantically segmented birds-eye-view map of the scene $\mathbf{Y}$. Formally, given an input image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ and its intrinsic matrix $\mathbf{C} \in \mathbb{R}^{3\times 3}$, our model predicts a set of binary variables $\mathbf{Y}^k \in \mathbb{R}^{X\times Z}$ for each class $k \in K$:

$$p(\mathbf{Y}^k|\mathbf{I},\mathbf{C}) = \mathbf{\Phi}(\mathbf{I},\mathbf{C}), \tag{1}$$

where Φ is a neural network trained to resolve both semantic and positional uncertainties.

The design of our network rests on our novel transformation between the image-plane $\mathbb{P}^I$ and BEV-plane $\mathbb{P}^{BEV}$. Our end-to-end approach, as shown in Fig. 1a, is composed of the following subtasks: (1) constructing representations in the image-plane which encode semantics and some knowledge of depth, (2) transforming the image-plane representation to BEV and (3) semantically segmenting the BEV-representation.

A. Image-to-BEV Translation

Transforming from image to BEV requires a mapping which determines the image pixel correspondence to BEV polar ray. As camera geometry dictates a 1-1 correspondence between each vertical scanline and its associated ray, we treat the mapping as a set of sequence-to-sequence translations. With reference to Fig. 1b, we want to find the discretized radial depths of elements in the vertical scan line of an image, up to r metres from the camera: we have an image column $S^I \in \mathbb{R}^H$, and we want to find its BEV ray $S^{\phi(BEV)} \in \mathbb{R}^r$, where H is the height of the column and r represents the radial distance from the camera. This mapping can be viewed as an assignment of semantic objects from the image-plane to their positional slots along a ray in the BEV-plane.

We propose learning the alignment between input scanlines and output polar rays through an attention mechanism [21]. We employ attention in two ways: (1) *inter-plane attention* as shown in Fig.1b, which initially assigns features from a scanline to a ray and (2) *polar ray self-attention* that globally reasons about its positional assignments across the ray. We motivate both uses below, starting with inter-plane attention.

Inter-plane attention: Consider a semantically segmented image column and its corresponding polar BEV ground truth. Here, alignment between the column and the ground truth ray is 'hard', *i.e.* each pixel in the polar ray corresponds to a single semantic category from the image column. Thus, the only uncertainty that must be resolved to make this a hard-assignment is the depth of each pixel. However, when making this assignment, we need to assign features that aid in resolving semantics and depth. Hence, a hard assignment would be detrimental. Instead, we want a soft-alignment, where every pixel in the polar ray is assigned a combination of elements in the image column, *i.e.* a *context* vector. Concretely, when generating each radial element $S_i^{\phi(BEV)}$, we want to give it a *context* c_i based on a convex combination of elements in the image column S^I and the radial position r_i of the element $S_i^{\phi(BEV)}$ along the polar ray. This need for context assignment motivates our use of soft-attention between the image column and its polar ray, as illustrated in Fig. 1.

Formally, let $\mathbf{h} \in \mathbb{R}^{H\times C}$ represent the encoded "memory" of an image column of height H, and let $\mathbf{y} \in \mathbb{R}^{r\times C}$ represent a *positional query* which encodes relative position along a polar ray of length r. We generate a context $\mathbf{c}$ based on the input sequence $\mathbf{h}$ and the query $\mathbf{y}$ through alignment $\boldsymbol{\alpha}$ between elements in the input sequence and their radial position. First, the input sequence $\mathbf{h}$ and positional query $\mathbf{y}$ are projected by matrices $W_Q \in \mathbb{R}^{C\times D}$ and $W_K \in \mathbb{R}^{C\times D}$ to the corresponding representations Q and K:

$$\begin{aligned} Q(\mathbf{y}_i) &= \mathbf{y}_i W_Q \\ K(\mathbf{h}_i) &= \mathbf{h}_i W_K. \end{aligned} \tag{2}$$

Following common terminology, we refer to Q and K as 'queries' and 'keys' respectively. After projection, an unnormalized alignment score $e_{i,j}$ is produced between each memory-query combination using the scaled-dot product [11]:

$$e_{i,j} = \frac{\langle Q(\mathbf{y}_i), K(\mathbf{h}_j)\rangle}{\sqrt{D}}. \tag{3}$$

The energy scalars are then normalized using a $\mathrm{softmax}$ to produce a probability distribution over the memory:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{H}\exp(e_{i,k})}. \tag{4}$$

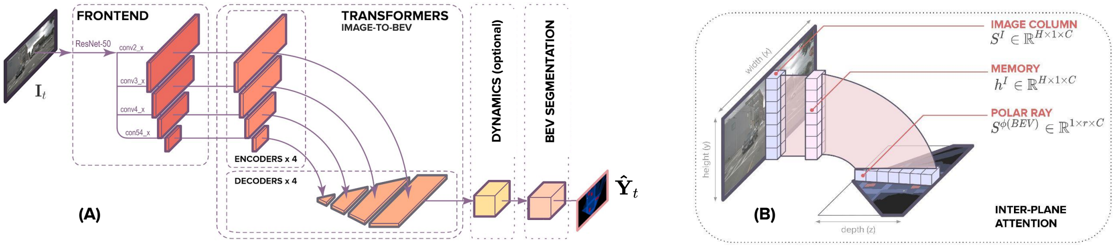

Fig. 1. (A) Our model architecture. The **Frontend** extracts spatial features at multiple scales. **Encoder-decoder transformers** translate spatial features from the image to BEV. An optional **Dynamics Module** uses past spatial BEV features to learn a spatiotemporal BEV representation. A **BEV Segmentation Network** processes the BEV representation to produce multi-scale occupancy grids. (B) Our inter-plane attention mechanism. In our attention-based model, vertical scan lines in the image are passed one by one to a transformer encoder to create a 'memory' representation which is decoded into a BEV polar ray.

Finally, the context vector is computed as a weighted sum of K:

$$c_i = \sum_{j=1}^{H} \alpha_{i,j} K(\mathbf{h}_j). \tag{5}$$

Generating the context this way allows each radial slot r_i to independently gather relevant information from the image column; and represents an initial assignment of components from the image to their BEV locations. Such an initial assignment is analogous to lifting a pixel based on its depth. However, it is lifted to a distribution of depths and thus should be able to overcome common pitfalls of sparsity and elongated object frustums. This means that the image-context available to each radial slot is decoupled from its distance to the camera. Finally, to generate BEV feature $S_i^{\phi(BEV)}$ at radial position r_i, we globally operate on the assigned contexts for *all* radial positions $\mathbf{c} = \{c_1, ..., c_r\}$:

$$S_i^{\phi(BEV)} = g(\mathbf{c}), \tag{6}$$

where $g(.)$ is a nonlinear function reasoning across the *entire* polar ray. We describe its role below.

Polar ray self-attention: The need for the non-linear function $g(.)$ as a global operator arises out of the limitations brought about by generating each context vector c_i independently. Given the absence of global reasoning for each context c_i, the spatial distribution of features across the ray is unlikely to be congruent with object shape, locally or globally. Rather, this distribution may only represent scattered suggestions of object-part positions. Therefore, we need to operate globally across the ray to allow the assigned scanline features to reason about their placement within the context of the entire ray, and thus aggregate information in a manner that generates coherent object shapes.

Global computation across the polar ray is computed much like soft-attention outlined in Eq. (2) - (5), except that the self-attention is applied to the ray only. Eq. (2) is recalculated with a new set of weight matrices with inputs to both equations replaced with the context vector c_i.

Extension to transformers: Our inter-plane attention can be extended to attention between the encoder-decoder of transformers by replacing the key $K(\mathbf{h}_j)$ in Eq. (5) with another projection of the memory $\mathbf{h}$, the 'value'. Similarly, polar-ray self-attention can be placed within a transformer-decoder by replacing the key in Eq. (5) with a projection of the context c_i to represent the value.

B. *Infinite lookback monotonic attention*

Although soft-attention is sufficient for learning an alignment between any arbitrary pair of source-target sequences, our sequences exist in the physical world where the alignment exhibits physical properties based on their spatial ordering. Typically, in urban environments, depth monotonically increases with height *i.e.,* as you move up the image, you move further away from the camera. We enforce this through monotonic attention with infinite lookback [25]. This constrains radial depth intervals to observe elements of the image column that are monotonically increasing in height but also allows context from the bottom of the column (or equivalently, previous memory entries).

Monotonic attention (MA) was originally proposed for computing alignments for simultaneous machine translation [26]. However, the 'hard' assignment between source and target sequence means important context is neglected. This led to the development of MA with infinite lookback (MAIL) [25], [27], [28], which combined hard MA with soft-attention that extends from the hard assignment to the beginning of the source sequence. We adopt MAIL as a way of constraining our attention mechanism to potentially prevent overfitting by ignoring the redundant context in the vertical scan line of an image. The primary objective of our adoption of MAIL is to understand whether context below a point in an image is more helpful than what is above.

We employ MAIL by first calculating a hard-alignment using monotonic attention. This makes a hard assignment of context c_i to an element of the memory $\mathbf{h}_j$, after which a soft-attention mechanism over previous memory entries $\mathbf{h}_1, ..., \mathbf{h}_{j-1}$ is applied. Formally, for each radial position $\mathbf{y}_i \in \mathbf{y}$ along the polar ray, the decoder begins scanning memory entries from index $j = t_{i-1}$, where t_i is the index of the memory entry chosen for position $\mathbf{y}_i$. For each memory entry, it produces a selection probability $p_{i,j}$, which corresponds to the probability of either stopping and setting $t_i = j$ and $c_i = \mathbf{h}_{t_i}$, or moving onto the next memory entry $j + 1$. As hard assignment is not differentiable, training is instead carried out with respect to the expected value of c_i, with the monotonic alignments $\alpha_{i,j}$ calculated as follows:

$$p_{i,j} = \text{sigmoid}(\text{Energy}(\mathbf{y}_i, \mathbf{h}_j)), \tag{7}$$

$$\alpha_{i,j} = p_{i,j}\left((1-p_{i,j-1})\frac{\alpha_{i,j-1}}{p_{i,j-1}} + \alpha_{i-1,j}\right), \quad (8)$$

where the Energy function is calculated in the same manner as Eq. (3). Assuming monotonic attention stops at t_i, the infinite lookback strategy first computes energies $e_{i,k}$ using equation Eq. 3 for $k \in 1, 2, ..., t_i$. The attention distribution over the allowed states is calculated as follows:

$$\beta_{i,j} = \sum_{k=j}^{H}\left(\frac{\alpha_{i,k}\exp(e_{i,k})}{\sum_{l=1}^{k}\exp(e_{i,l})}\right). \quad (9)$$

This effectively represents a distribution over image-elements which lie below a point in the image; to calculate a distribution over only what lies above a point in the image, the image column can be flipped. The context vector is calculated similar to inter-plane attention, where $c_i = \sum_{j=1}^{H}\beta_{i,j}K(\mathbf{h}_j)$.

C. Model architecture

We build an architecture that facilitates our goal of predicting a semantic BEV map from a monocular image around this alignment model. As shown in Fig. 1, it contains three main components: a standard CNN backbone which extracts spatial features in the image-plane, encoder-decoder transformers to translate features from the image-plane to BEV and finally a segmentation network which decodes BEV features into semantic maps.

2D Multi-scale feature learning in $\mathbb{P}^I$: Reconstructing an image in BEV requires representations which can detect scene elements at varying depths and scale. Like prior object detection methods [8], [10], [29], we handle this scale variance using a CNN backbone with a feature pyramid to produce feature maps $\mathbf{f}^I_{t,s} \in \mathbb{R}^{C\times h_s \times w_s}$ at multiple scales $u \in U$.

1D Transformer encoders in $\mathbb{P}^I$: This component encodes long-range vertical dependencies across the input features through self-attention, using an encoder for each scale u of features (second left block of Fig. 1a). Each scale of features $\mathbf{f}^I_{t,u}$ is first reshaped into its individual columns, creating w_u sequences of length h_u and dimension C. Each encoder layer has a standard architecture consisting of multi-head attention and a feed forward network. Given the permutation invariance of the transformer, we add fixed 1D sinusoid positional encodings [11] to the input of each attention layer. The U encoders each produce a memory $\mathbf{h}^I_{t,u} \in \mathbb{R}^{w_u \times h_u \times C}$.

1D Transformer decoders in $\mathbb{P}^{BEV}$: This component generates independent sequences of BEV features along a polar ray through multi-head attention across the encoder memory. As shown in the second left block of Fig. 1, there is one transformer decoder for each transformer encoder. Every encoded image column $\mathbf{h}^I \in \mathbb{R}^{h_u \times C}$ is transformed to a BEV polar ray $\mathbf{f}^{\phi(BEV)} \in \mathbb{R}^{r_u \times C}$, where r_u is the radial distance along the ray. Given the desired output sequence of length r_u, the decoder takes in r_u positional embeddings, which we refer to as *positional queries*. These are r_u unique embeddings with fixed sinusoid positional information added to them, just like our encoder above. When replacing the encoder-decoder multi-head soft-attention with monotonic attention, each head in the decoder is replaced with a monotonic attention head from Eq. (8). The U decoders each output w_u BEV sequences of length r_u along the polar ray, producing a polar encoding $\mathbf{f}^{\phi(BEV)} \in \mathbb{R}^{w_u \times r_u \times C}$. Similar to prior work which builds stixel representations from an image [30], [31], each image column in our model corresponds to an angular coordinate in the polar map. Finally we concatenate along the ray to obtain a single 2D polar feature map and convert to a rectilinear grid, to create our BEV representation $\mathbf{f}^{BEV}_t \in \mathbb{R}^{C\times Z \times X}$.

Our transformer encoder and decoder use the same set of projection matrices for every sequence-to-sequence translation, giving it a structure that is convolutional along the x-axis and allowing us to make efficient use of data when training. We constrain our translations to 1D sequences as opposed to using the entire image to make learning easier, a decision we analyze in section IV-A.

Polar-adaptive context assignment: The positional encodings applied to the transformer so far have all been 1D. While this allows our convolutional transformer to leverage spatial relationships between height in the image and depth, it remains agnostic to polar angle. However, the angular domain plays an important role in urban environments. For instance, images display a broadly structured distribution of object classes across their width (e.g. pedestrians are typically only seen on sidewalks, which lie towards the edges of the image). Furthermore, object appearance is also structured along the width of the image as they are typically orientated along orthogonal axes and viewing angle changes its appearance. To account for such variations in appearance and distribution across the image, we add additional positional information by encoding polar angle in our 1D scanline-to-ray translations.

Dynamics with axial attention in $\mathbb{P}^{BEV}$: This component incorporates temporal information from past estimates to build a spatiotemporal BEV representation of the present. As the representations built by the previous components are entirely spatial, we add a simple component based on axial attention to make the model temporally aware. The placement of this optional module can be seen in Fig. 1a. We obtain BEV features for multiple timesteps, creating a representation $\mathbf{f}^{BEV}_{1:t} \in \mathbb{R}^{T\times C\times Z\times X}$. We apply axial-attention across the spatial and temporal axes, giving every pixel at every timestep axial context from the other timesteps. Our temporal aggregation means the features of any timestep now contain dynamics across the sequence, and the module can use any of these features in its forward pass. This module is optional as it builds a spatiotemporal representation. It can be omitted when constructing a purely spatial model.

Segmentation in $\mathbb{P}^{BEV}$: To decode our BEV features into semantic occupancy grids, we adopt a convolutional encoder-decoder structure used in prior segmentation networks [10], [32]. The aggregated module structure (right block of Fig. 1a), takes BEV features $\mathbf{f}^{BEV}_t \in \mathbb{R}^{C\times Z\times X}$ and outputs occupancy grids $\mathbf{m}^{BEV}_{t,u} \in \mathbb{R}^{classes\times x_u \times z_u}$ for scales $u \in U$. Moving from the 1D attention mechanisms of our transformer to the two-dimensional locality of convolutions provides contextual reasoning across the horizontal x-axis which helps stitch together potential discontinuities between adjacent polar rays and their subsequent rectilinear resampling.

Loss in $\mathbb{P}^{BEV}$: As the training signal provided to the predicted occupancy grids must resolve both semantic and positional uncertainties, we use the same multi-scale Dice loss as [10]. At each scale u, the mean Dice Loss across classes K is:

$$\mathcal{L}^{u} = 1 - \frac{1}{|K|}\sum_{k=1}^{K} \frac{2\sum_{i}^{N} \hat{y}_i^k y_i^k}{\sum_{i}^{N} \hat{y}_i^k + y_i^k + \epsilon}, \tag{10}$$

where y_i^k is the ground truth binary variable grid cell, $\hat{y}_i^k$ the predicted sigmoid output of the network, and ϵ is a constant used to prevent division by zero.

IV. Experiments and Results

We evaluate the effectiveness of treating the image-to-BEV transformation as a translation problem on the nuScenes dataset [33]; with ablations on lookback direction in monotonic attention, the utility of long-range horizontal context and the effect of polar positional information. Finally, we compare our approach to current state-of-the-art approaches on the **nuScenes [33], Argoverse [34]** and **Lyft [35]** datasets.

Dataset: The nuScenes dataset [33] consists of 1000 20-second clips captured across Boston and Singapore, annotated with 3D bounding boxes and vectorized road maps. We follow [8]'s data generation process, object classes and training/validation splits to allow fair comparison. We use nuScenes for our ablation studies as it is considerably larger and contains more object categories.

Implementation: Our frontend uses a pretrained ResNet-50 [36] with a feature pyramid [37] on top. BEV feature maps built by the transformer decoder have a resolution of 100×100 pixels, with each pixel representing 0.5m^2 in the world. Our spatiotemporal model takes a 6Hz sequence of 4 images, where the final frame is the time step we make the prediction for. Our largest scale output is 100×100 pixels, which we upsample to 200×200 for fair evaluation with the literature. We train our network end-to-end with an Adam optimizer, batch size 8 and initial learning rate of $5e{-}5$, which we decay by 0.99 every epoch for 40 epochs.

A. Ablation studies

Which way to look? In Table II (top) we compare soft-attention (looking both ways), monotonic attention with lookback towards the bottom of the image (looking down) and monotonic attention with lookback towards the top of the image (looking up). The results indicate looking downwards from a point in the image is better than looking upwards. This is consistent with how humans try to determine the distance of an object in an urban environment — along with local textural clues of scale, we make use of where the object intersects the ground plane. The results also show that looking in both directions further increases accuracy, making it more discriminative for depth reasoning.

Long-range horizontal dependencies: As our image-to-BEV transformation is carried out as a set of 1D sequence-to-sequence translations, a natural question is what happens when the entire image is translated to BEV (similar to 'lift' approaches [9], [20]). Given the quadratic computation time and memory required to produce attention maps, this is prohibitively expensive. However, we can approximate the contextual benefits of using the entire image by applying horizontal axial-attention on the image-plane features before the transformation. With axial-attention across the rows of the image, the pixels in the vertical scanline now have long-range horizontal context, after which we provide long-range vertical context as before by translating between 1D sequences.

Table II (middle) shows that incorporating long-range horizontal context does not benefit the model and its impact is slightly detrimental. This suggests two things. Firstly, every transformed ray does not need information from the entire width of the input image, or rather, the long-range context does not provide any additional benefit over the context that has already been aggregated through the convolutions of our frontend. This indicates that performing the translation using the entire image would not increase model accuracy over the constrained formulation of our baseline. Finally, the decrease in performance from the introduction of horizontal axial-attention is possibly a sign of the difficulty in training using attention for sequences which are the width of the image; we should expect that using the entire image as the input sequence would be much harder to train.

Polar-agnostic vs polar-adaptive transformers: Table II (bottom) compares a polar-agnostic (Po-Ag) transformer to its polar-adaptive (Po-Ad) variants. A Po-Ag model has no polar-positional information, Po-Ad in the image-plane involves polar encodings added to the transformer encoder while for the BEV-plane this information is added to the decoder. Adding polar encodings to any one plane provides similar benefit over an agnostic model, with dynamic classes increasing the most. Adding it to both planes increases this further, but has the largest impact on static classes.

B. Comparison to state-of-the-art

Baselines: We compare against a number of prior state-of-the-art methods. We begin our comparison against 'compression' approaches [8], [10] on nuScenes and Argoverse using the train/val splits of [8]. We then compare against the 'lift' approach of [9], [20] on nuScenes and Lyft.

In Table I, our spatial model outperforms the current state-of-the-art compression approach of STA-S [10] with a mean relative improvement 15%. It is the smaller dynamic classes in particular on which we show significant improvement, with buses, trucks, trailers and barriers all increasing by a relative 35-45%. This is supported by our qualitative results in Fig. 2, where our models show greater structural similarity to the ground truths and a better sense of shape. This difference can be partly attributed to the fully-connected layer (FCL) used in compression: when detecting small, distant objects, a large portion of the image is redundant context. Expecting the weights of the FCL to ignore redundancies to maintain only the small objects in the bottleneck is a challenge. Furthermore, objects such as pedestrians are often partially occluded by vehicles. In such cases, the FCL would be inclined to ignore the pedestrian and instead maintain the vehicle's semantics. Here the attention method shows its advantages as each radial

TABLE I

IoU(%) on the nuScenes validation split and baseline results of [8].

Method	Drivable	Crossing	Walkway	Carpark	Bus	Bike	Car	Cons.Veh.	Motorbike	Trailer	Truck	Ped.	Cone	Barrier	Mean
VED [6]	54.7	12.0	20.7	13.5	0.0	0.0	8.8	0.0	0.0	7.4	0.2	0.0	0	4.0	8.7
VPN [2]	58.0	27.3	29.4	12.3	20.0	4.4	25.5	4.9	5.6	**16.6**	17.3	7.1	4.6	10.8	17.5
PON [8]	60.4	28.0	31.0	18.4	20.8	9.4	24.7	12.3	7.0	**16.6**	16.3	8.2	5.7	8.1	19.1
STA-S [10]	71.1	31.5	32.0	28.0	22.8	14.6	34.6	10.0	7.1	11.4	18.1	7.4	5.8	10.8	21.8
Our Spatial	**72.6**	**36.3**	**32.4**	**30.5**	**32.5**	**15.1**	**37.4**	**13.8**	**8.1**	15.5	**24.5**	**8.7**	**7.4**	**15.1**	**25.0**
STA-ST [10]	70.7	31.1	32.4	**33.5**	29.2	12.1	36.0	12.1	**8.0**	13.6	22.8	8.6	6.9	14.2	23.7
Our Spatiotemp.	**74.5**	**36.6**	**35.9**	31.3	**32.8**	**14.7**	**39.7**	**14.2**	7.6	**13.9**	**26.3**	**9.5**	**7.6**	**14.7**	**25.7**

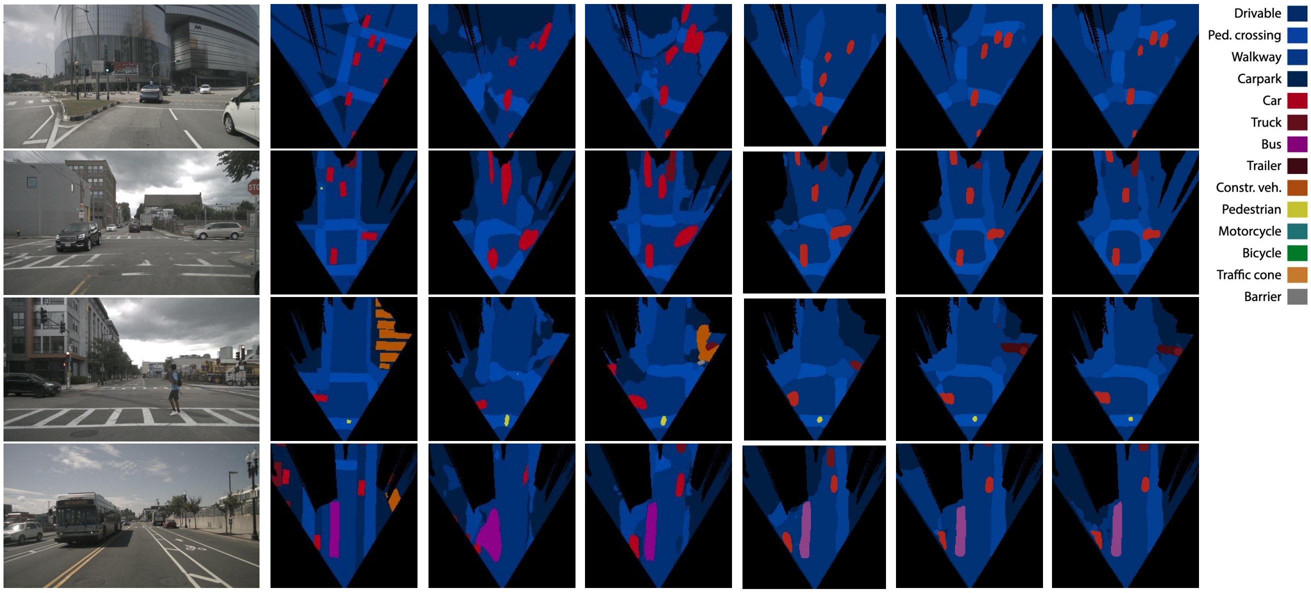

Fig. 2. Qualitative results on the nuScenes validation set of [8]. We compare against baseline results of prior work reported in [8] and follow their colour scheme. For fair comparison, we apply the ground truth visibility mask (black) to the predicted images as was done in [8].

TABLE II

IoU(%) for ablation studies.

Model	Static classes	Dynamic classes	Mean
Looking down	29.5	15.8	22.1
Looking up	29.9	17.1	23.0
Looking both ways	**32.4**	**19.4**	**25.0**
Baseline	**32.4**	**19.4**	**25.0**
Baseline w/ h. context	29.4	17.3	22.9
Po-Ag	30.3	18.1	23.7
Po-Ad (image-plane)	30.9	19.1	24.2
Po-Ad (BEV-plane)	31.3	19.2	24.3
Po-Ad (both planes)	**32.4**	**19.4**	**25.0**

depth can independently attend to the image — so the further depths can look at the pedestrian's visible body, while depths before can attend to the vehicle. Our results on the Argoverse dataset in Table III demonstrate similar patterns, where we improve upon PON [8] by a relative 30%.

In Table IV we outperform LSS [9] and FIERY [20] on nuScenes and Lyft (FIERY [20] uses the 'lift' approach of [9]). A true comparison on Lyft is not possible as it doesn't have a canonical train/val split and we were unable to acquire those used by [9]. While we used splits of similar sizes to [9], the exact scenes are unknown. As a 'lift' approach bears some similarity to our translation approach in that the network is able to select how to distribute image context across its polar ray, the difference in performance here can likely be attributed to our constrained, spatially-aware translations between scanlines and rays. One of the avenues for future work is improving localisation accuracy for distant objects, and their false negatives. Finally, our approach is easily transferrable to indoor mobile robotics applications once ground truth has been collected to train the models.

TABLE III

IoU(%) on the Argoverse validation split of [8].

	Driv.	Veh.	Ped.	L.Veh.	Bic.	Bus.	Trail.	Mot.	Mean
PON [8]	65.4	31.4	**7.4**	11.1	3.6	11	0.7	5.7	17.0
Ours	**75.9**	**35.8**	5.7	**14.9**	**3.7**	**30.2**	**12.2**	2.6	**22.6**

TABLE IV

IoU(%) against spatial (S) and spatiotemporal (ST) methods on nuScenes and Lyft.

	nuScenes			Lyft		
	Driv.	Car	Veh.	Driv.	Car	Veh.
(S) LSS	72.9	32.0	32.0	-	43.1	44.6
(S) FIERY	-	37.7	-	-	-	-
(S) Ours	**78.9**	**39.9**	**38.9**	**82.0**	**45.9**	**45.4**
(ST) FIERY	-	39.9	-			
(ST) Ours	**80.5**	**41.3**	**40.2**			

V. Conclusion

We proposed a novel use of transformer networks to map from images and video sequences to an overhead map or bird's-eye-view of the world. We combine our physical-grounded and constrained formulation, with ablation studies that make use of progress in monotonic attention to confirm our intuitions whether context above or below a point is more important for this form of map generation. Our novel formulation obtains state-of-the-art results for instantaneous mapping of three well-established datasets.

References

[1] S. Sengupta, P. Sturgess, L. Ladickỳ, and P. H. Torr, "Automatic dense visual semantic mapping from street-level imagery," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 857–862.

[2] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, 2020.

[3] B. Liu, B. Zhuang, S. Schulter, P. Ji, and M. Chandraker, "Understanding road layout from videos as a whole," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4414–4423.

[4] Z. Wang, B. Liu, S. Schulter, and M. Chandraker, "A parametric top-view representation of complex road scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 325–10 333.

[5] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 787–802.

[6] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 445–452, 2019.

[7] K. Mani, S. Daga, S. Garg, S. S. Narasimhan, M. Krishna, and K. M. Jatavallabhula, "Monolayout: Amodal scene layout from a single image," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1689–1697.

[8] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Proceedings of the European Conference on Computer Vision*, 2020.

[10] A. Saha, O. Mendez, C. Russell, and R. Bowden, "Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation," in *Proceedings of the International Conference on Robotics and Automation*, 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] D. Wang, C. Devin, Q.-Z. Cai, P. Krähenbühl, and T. Darrell, "Monocular plan view networks for autonomous driving," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2876–2883.

[13] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.

[14] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[15] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1991–1999.

[16] P. Poirson, P. Ammirato, C.-Y. Fu, W. Liu, J. Kosecka, and A. C. Berg, "Fast single shot detection and pose estimation," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 676–684.

[17] A. Palazzi, G. Borghi, D. Abati, S. Calderara, and R. Cucchiara, "Learning to map vehicles into bird's eye view," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 233–243.

[18] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.

[19] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[20] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "FIERY: Future instance segmentation in bird's-eye view from surround monocular cameras," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*, 2019.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners."

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision*, 2020.

[25] N. Arivazhagan, C. Cherry, W. Macherey, C.-C. Chiu, S. Yavuz, R. Pang, W. Li, and C. Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1313–1323.

[26] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *International Conference on Machine Learning*, 2017, pp. 2837–2846.

[27] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *International Conference on Learning Representations*, 2018.

[28] X. Ma, J. M. Pino, J. Cross, L. Puzon, and J. Gu, "Monotonic multihead attention," in *International Conference on Learning Representations*, 2019.

[29] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[30] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world-a compact medium level representation of the 3d-world," in *Joint Pattern Recognition Symposium*. Springer, 2009, pp. 51–60.

[31] D. Pfeiffer and U. Franke, "Modeling dynamic 3d environments by means of the stixel world," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 3, pp. 24–36, 2011.

[32] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.

[33] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[34] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[35] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska *et al.*, "Lyft level 5 av dataset 2019," *urlhttps://level5. lyft. com/dataset*, 2019.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.