A Architecture

In total, our architecture has 30.4M parameters, comprising of modules:

- Perception, $E_{\text{perception}}$, 25.3M parameters ;
- Dynamics, $\mathcal{Y}$, and present/future distributions, $\mathcal{P}$ and $\mathcal{F}$, 0.8M parameters ;
- Future prediction, $\mathcal{G}$, 3.5M parameters ;
- Control policy model, $\mathcal{C}$, 0.7M parameters.

A.1 Perception

Semantics and Geometry. Our model is an encoder-decoder model with five encoder blocks and three decoder blocks, followed by an atrous spatial pyramid pooling (ASPP) module [9]. The encoders contain 2, 4, 8, 8, 8 layers respectively, downsampling by a factor of two each time with a strided convolution. The decoders contain 3 layers each, upsampling each time by a factor of two with a sub-strided convolution. All layers have residual connections and many are low rank, with varying kernel and dilation sizes. Furthermore, we employ skip connections from the encoder to decoder at each spatial scale.

We pretrain the scene understanding encoder on a number of heterogeneous datasets to predict semantic segmentation and depth: CityScapes [14], Mapillary Vistas [48], ApolloScape [29] and Berkeley Deep Drive [67]. We collapse the classes to 14 semantic segmentation classes shared across these datasets and sample each dataset equally during training. We train for 200,000 gradient steps with a batch size of 32 using SGD with an initial learning rate of 0.1 with momentum 0.9. We use cross entropy for segmentation and the scale-invariant loss [44] to learn depth with a weight of 1.0 and 0.1, respectively.

Motion. In addition to this semantics and geometry encoder, we also use a pretrained optical flow network, PWCNet [60]. We use the pretrained authors' implementation.

Perception. To form our perception encoder we concatenate these two feature representations (from the perception encoder and optical flow net) concatenated together. We use the features two layers before the output optical flow regression as the feature representation. The decoders of these networks are used for generating pseudo-ground truth segmentation and depth labels to train our dynamics and future prediction modules.

A.2 Training

Our model was trained on 8 GPUs, each with a batch size of 4, for 200,000 steps using an Adam optimiser with learning rate 3e−4. The input of our model is a sequence of 15 frames at resolution 224×480 and a frame rate of 5Hz (256×512 and 17Hz for Cityscapes). The first 5 frames correspond to the past and present context (1s), and the following 10 frames to the future we want to predict (2s). All layers in the network use batch normalisation and a ReLU activation function. We now describe each module of our architecture in more detail.

Dynamics four temporal blocks with kernel size $k = (2, 3, 3)$, stride $s = 1$ and output channels $c = [80, 88, 96, 104]$. In between every temporal block, four 2D residual convolutions ($k = (3, 3)$, $s = 1$) are inserted.

Present and Future Distribution two downsampling 2D residual convolutions ($k = (3, 3)$, $s = 2$, $c = [52, 52]$). An average pooling layer flattens the feature spatially, and a final dense layer maps it to a vector of size $2L$ ($L = 16$).

Future Prediction the main structure is a convolutional GRU ($k = (3, 3)$, $s = 1$). Each convolutional GRU is followed by three 2D residual convolutions ($k = (3, 3)$, $s = 1$). This structure is stacked five times. The decoders: two upsampling convolutions ($k = (3, 3)$, $s = 1$, $c = 32$), a convolution ($k = (3, 3)$, $s = 1$, $c = 16$), and finally a convolution without activation followed by a bilinear interpolation to the original resolution 224×480.

Control two downsampling convolutions ($k = (3, 3)$, $s = 2$, $c = [64, 32]$), followed by dense layers ($c = [1024, 512, 256, 128, 64, 32, 16, 4]$).

A.3 Temporal Block

We ablate the architecture of our proposed Temporal Block module on Cityscapes by evaluating performance of future semantic segmentation prediction, at resolution 256×512 and for future frames 5 and 10. Let k_t denote the temporal kernel size and k_s the spatial kernel size of the 3D convolutions. We compare the following modules:

(i) (k_t, k_s, k_s) and $(1, k_s, k_s)$ convolutions. No global context.
(ii) (k_t, k_s, k_s), $(k_t, 1, k_s)$ and $(1, k_s, k_s)$ convolutions. No global context.
(iii) (k_t, k_s, k_s), $(k_t, k_s, 1)$ and $(1, k_s, k_s)$ convolutions. No global context.
(iv) (k_t, k_s, k_s), $(k_t, 1, k_s)$, $(k_t, k_s, 1)$ and $(1, k_s, k_s)$ convolutions. No global context.
(v) (k_t, k_s, k_s), $(k_t, 1, k_s)$, $(k_t, k_s, 1)$ and $(1, k_s, k_s)$ convolutions. With global context (*i.e.* our proposed Temporal Block).

	Temporal Model	$\text{IoU}_{i=5}$ (↑)	$\text{IoU}_{i=10}$ (↑)
	Repeat frame	0.393	0.331
Probabilistic	(i) (k_t, k_s, k_s)	0.454	0.411
	(ii) (k_t, k_s, k_s), $(k_t, 1, k_s)$	0.461	0.411
	(iii) (k_t, k_s, k_s), $(k_t, k_s, 1)$	0.449	0.413
	(iv) (k_t, k_s, k_s), $(k_t, 1, k_s)$, $(k_t, k_s, 1)$	0.453	0.413
	(v) **Temporal Block (Ours)**	**0.464**	**0.416**

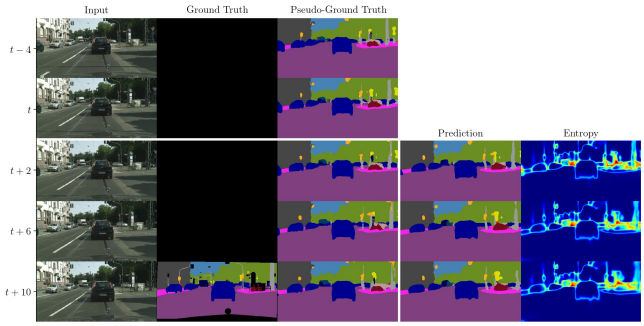
Table 5: Ablation study of the Temporal Block on Cityscapes, evaluated on future semantic segmentation performance at $i = 5$ and $i = 10$ frames in the future. Our proposed Temporal Block module outperforms all the other variants.

B Nomenclature

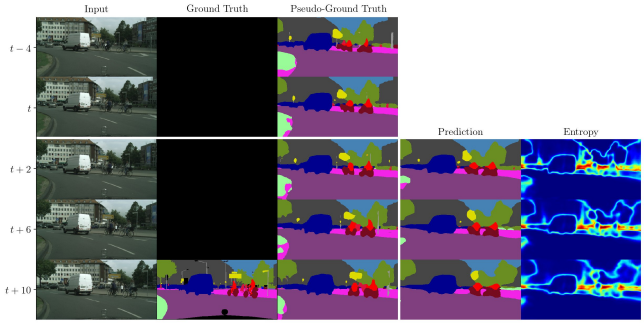
We detail the symbols used to describe our model in this paper.

Networks	
Perception encoder	$E_{\text{perception}}$
Temporal Block	$\mathcal{T}$
Dynamics module	$\mathcal{Y}$
Present network	$\mathcal{P}$
Future network	$\mathcal{F}$
Future prediction module	$\mathcal{G}$
Future decoders	$\mathcal{D}_s, \mathcal{D}_d, \mathcal{D}_f$
Control module	$\mathcal{C}$
Tensors	
Temporal context	T
Future prediction horizon	N_f
Future control horizon	N_c
Input image	i_t
Perception features	$x_t = E_{\text{perception}}(i_t)$
Dynamics features	$z_t = \mathcal{Y}(x_{t-T+1} : x_t)$
Present distribution	$\mu_{t,\text{present}}, \sigma_{t,\text{present}} = \mathcal{P}(z_t)$
Future distribution	$\mu_{t,\text{future}}, \sigma_{t,\text{future}} = \mathcal{F}(z_t)$
Noise vector (train)	$\eta_t \sim \mathcal{N}(\mu_{t,\text{future}}, \sigma^2_{t,\text{future}})$
Noise vector (test)	$\eta_t \sim \mathcal{N}(\mu_{t,\text{present}}, \sigma^2_{t,\text{present}})$
Future prediction inputs	$u_t^{t+i} = \eta_t$
Future prediction initial hidden state	$g_t^t = z_t$
Future prediction output features	$g_t^{t+i} = \mathcal{G}(u_t^{t+i}, g_t^{t+i-1})$
Future perception outputs	$\hat{o}_t^{t+i} = \{\hat{s}_t^{t+i}, \hat{d}_t^{t+i}, \hat{f}_t^{t+i}\}$ $= \{\mathcal{D}_s(g_t^{t+i}), \mathcal{D}_d(g_t^{t+i}), \mathcal{D}_f(g_t^{t+i})\}$
Control outputs	$\hat{c}_t = \{\hat{v}_t, \hat{\dot{v}}_t, \hat{\theta}_t, \hat{\dot{\theta}}_t\}$ $= \mathcal{C}(z_t)$

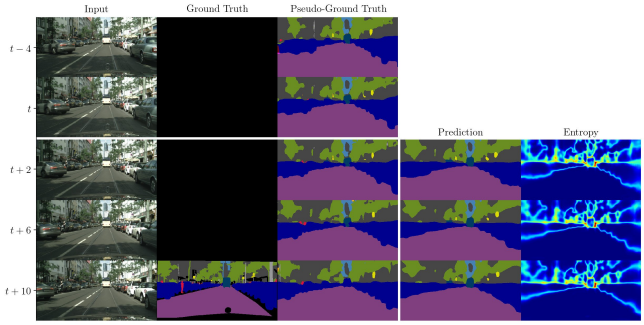
C Cityscapes Qualitative Examples



(a) Our model can correctly predict future segmentation of small classes such as poles or traffic lights.



(b) Dynamic agents, *i.e.* cars and cyclists, are also accurately predicted.



(c) In this example, the bus is correctly segmented, without any class bleeding contrary to the pseudo-ground truth segmentation, showing that our model can reason in a holistic way.

Fig. 5: Future prediction on the CityScapes dataset, for 10 frames in the future at 17Hz and 256×512 resolution.