A Additional Results

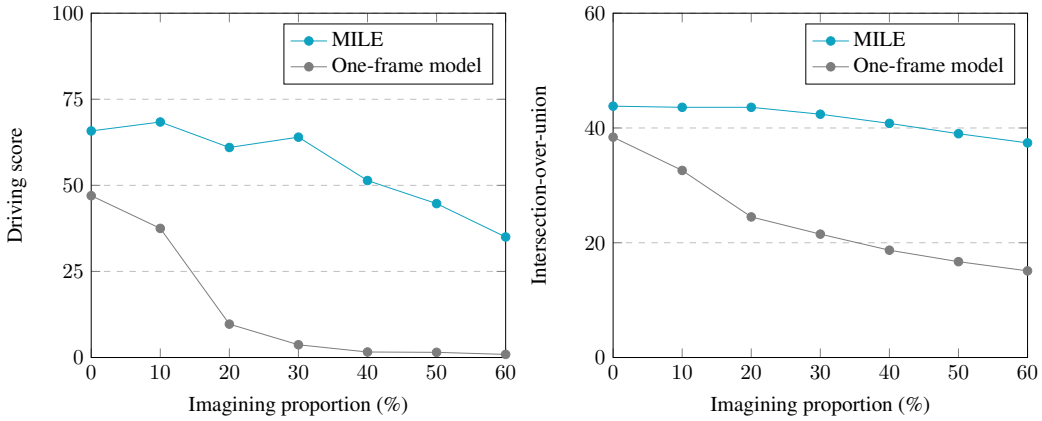
A.1 Driving in Imagination

We deploy the model in the fully recurrent setting, and at fixed intervals: (i) we let the model imagine future states and actions, without observing new images, and execute those actions in the simulator. (ii) We then let the model update its knowledge of the world by observing new image frames. More precisely, we set the fixed interval to a two-second window, and set a ratio of imagining vs. observing. If for example that ratio is set to 0.5, we make the model imagine by sampling from the prior distribution for 1.0s, then sample from the posterior distribution for 1.0s, and alternate between these two settings during the whole evaluation run.

We make the ratio of imagining vs. observing vary from 0 (always observing each image frame, which is the default behaviour) to 0.6 (imagining for 60% of the time). We report both the driving performance and perception accuracy in Figure 4. The driving performance is measured with the driving score, and the perception accuracy using the intersection-over-union with the ground truth BeV semantic segmentation. We compare MILE with a one-frame baseline which has no memory and only uses a single image frame for inference.

Figure 4a shows that our model can imagine for up to 30% of the time without any significant drop in driving performance. After this point, the driving score starts decreasing but remains much higher than its one-frame counterpart. In Figure 4b, we see that the predicted states remain fairly accurate (by decoding to BeV segmentation), even with an important amount of imagining. These results demonstrate that our model can predict plausible future states and actions, accurate enough to control a vehicle in closed-loop.

Figure 5 illustrates an example of the model driving in imagination and successfully negotiating a roundabout.



(a) Driving performance.

(b) Perception accuracy.

Figure 4: Driving in imagination. We report the closed-loop driving performance and perception accuracy in CARLA when the model imagines future states and actions and does not observe a proportion of the images.

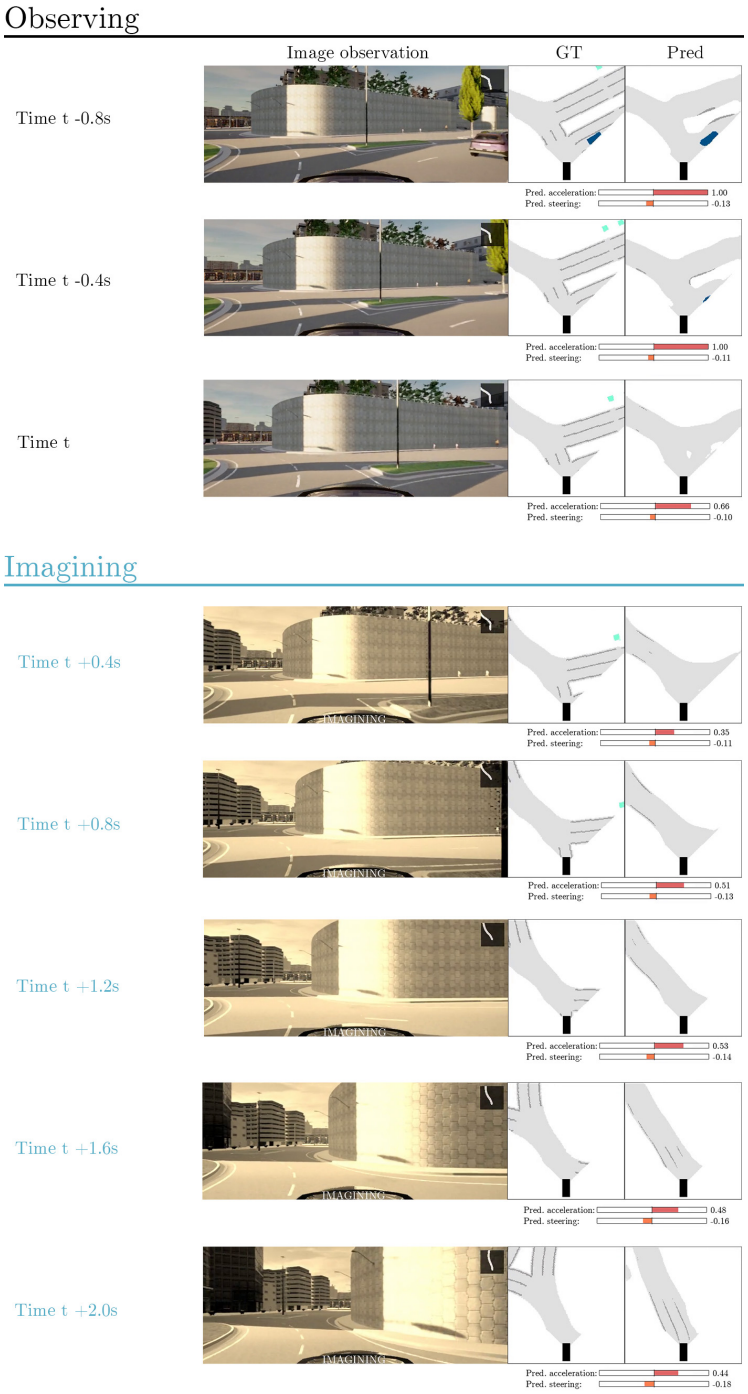


Figure 5: An example of the model imagining and accurately predicting future states and actions to negotiate a roundabout. When imagining, the model does not observe the image frames, but predicts the future states and actions from its current latent state.

A.2 Image Resolution

In urban driving, small elements in the scene can have an important role in decision making. One typical example is traffic lights, which only occupy a small portion of the image, but dictate whether a vehicle can continue driving forward or needs to stop at a red light. Figure 6 and Figure 7 illustrate how traffic lights and pedestrians become much harder to distinguish in lower image resolutions.

We evaluate the importance of image resolution by training MILE at different resolutions: 75×120, 150×240, 300×480, and 600×960 (our proposed resolution). We report the results in Table 4 and observe a significant decrease in both driving score and cumulative reward. The performance drop is most severe in the infraction penalty metric. To get a better understanding of what is happening, we detail in Table 5 the breakdown of the infractions. We report the number of red lights run, the number of vehicle collisions, and the number of pedestrian collisions, all per kilometre driven. As the resolution of the image lowers, the number of infractions increases across all modalities (red lights, vehicles, and pedestrians). These results highlight the importance of high resolution images to reliably detect traffic lights, vehicles, and pedestrians.

Table 4: Analysis on the image resolution. We report driving performance on a new town and new weather conditions in CARLA.

Image resolution	Driving Score	Route	Infraction	Reward	Norm. Reward
75×120	20.9 ± 0.0	87.5 ± 0.0	25.3 ± 0.0	5674 ± 0.0	0.65 ± 0.0
150×240	27.9 ± 0.0	81.8 ± 0.0	40.4 ± 0.0	5017 ± 0.0	0.65 ± 0.0
300×480	43.3 ± 0.0	96.1 ± 0.0	44.4 ± 0.0	5814 ± 0.0	0.55 ± 0.0
600×960	$\mathbf{61.1 \pm 3.2}$	$\mathbf{97.4 \pm 0.8}$	$\mathbf{63.0 \pm 3.0}$	$\mathbf{7621 \pm 460}$	$\mathbf{0.67 \pm 0.02}$
Expert	88.4 ± 0.9	97.6 ± 1.2	90.5 ± 1.2	8694 ± 88	0.70 ± 0.01

Table 5: Analysis on the image resolution. We report the breakdown of infraction penalties on a new town and new weather conditions in CARLA. The metrics are: number of red lights run, number of vehicle collisions, and number of pedestrian collisions. They are normalised per kilometre driven. Lower is better.

Image resolution	Red lights (↓)	Vehicles (↓)	Pedestrians (↓)
75×120	3.07 ± 0.0	0.77 ± 0.0	0.07 ± 0.0
150×240	2.39 ± 0.0	0.35 ± 0.0	0.03 ± 0.0
300×480	0.99 ± 0.0	0.31 ± 0.0	0.05 ± 0.0
600×960	$\mathbf{0.13 \pm 0.04}$	$\mathbf{0.24 \pm 0.05}$	$\mathbf{0.01 \pm 0.01}$
Expert	0.04 ± 0.01	0.15 ± 0.01	0.02 ± 0.00

A.3 Training Town Evaluation

We also evaluate our method on towns and weather conditions seen during training. As reported in Table 6, our model shows a 21% relative improvement in driving score with respect to Roach. Note that the RL expert has a lower performance than in test town Town05, because Town03 was designed as the most complex town [52].

Table 6: Driving performance in CARLA on a town and weather conditions seen during training. Metrics are averaged across three runs.

	Driving Score	Route	Infraction	Reward	Norm. Reward
Roach [9]	50.6 ± 1.9	$\mathbf{91.0 \pm 0.7}$	56.9 ± 1.2	4419 ± 487	0.38 ± 0.04
MILE	$\mathbf{61.4 \pm 0.3}$	89.3 ± 2.5	$\mathbf{69.4 \pm 1.3}$	$\mathbf{7627 \pm 190}$	$\mathbf{0.71 \pm 0.01}$
Expert	81.5 ± 2.8	95.1 ± 1.2	85.6 ± 1.7	7740 ± 220	0.69 ± 0.03

(a) Resolution 600×960.

(b) Resolution 300×480.

(c) Resolution 150×240.

(d) Resolution 75×120.

Figure 6: Input image observation at different resolutions. The red traffic light becomes almost indistinguishable in lower resolutions.

A.4 Evaluation in the Settings of Past Works

We also evaluated our model in the evaluation settings of:

- TransFuser [48]: the full 10 test routes of Town05 in ClearNoon weather and no scenarios (Table 7);
- LAV [10]: 2 test routes from Town02 and 2 test routes in Town05 in weathers [SoftRainSunset, WetSunset, CloudyNoon, MidRainSunset] and no scenarios (Table 8).

Table 7: Driving performance in CARLA in the TransFuser [48] evaluation setting.

	Driving Score	**Route**	**Infraction**	**Reward**	**Norm. Reward**
TransFuser [48]	43.7 ± 2.4	79.6 ± 8.5	-	-	-
MILE	$\mathbf{69.9} \pm \mathbf{7.0}$	$\mathbf{98.3} \pm \mathbf{2.1}$	70.9 ± 6.8	7792 ± 663	0.69 ± 0.03

Table 8: Driving performance in CARLA in the LAV [10] evaluation setting.

	Driving Score	**Route**	**Infraction**	**Reward**	**Norm. Reward**
LAV [10]	54.2 ± 8.0	78.7 ± 5.8	$\mathbf{73.0} \pm \mathbf{4.9}$	-	-
MILE	$\mathbf{64.3} \pm \mathbf{5.2}$	$\mathbf{99.1} \pm \mathbf{1.5}$	64.6 ± 5.4	9631 ± 341	0.72 ± 0.01

(a) Resolution 600×960.

(b) Resolution 300×480.

(c) Resolution 150×240.

(d) Resolution 75×120.

Figure 7: Input image observation at different resolutions. It becomes increasingly harder to see the pedestrian as the resolution decreases.

B Lower Bound Derivation

Proof. Let $q_{H,S} \triangleq q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) = q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1})$ be the variational distribution (where we have assumed independence of $(\mathbf{y}_{1:T}, \mathbf{a}_T)$ given $(\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1})$), and $p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T})$ be the posterior distribution. The Kullback-Leibler divergence between these two distributions writes as:

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T})\right) \\
&= \mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log \frac{q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T})}{p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T})}\right] \\
&= \mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log \frac{q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T})}{p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}) p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}|\mathbf{h}_{1:T}, \mathbf{s}_{1:T})}\right] \\
&= \log p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) - \mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}|\mathbf{h}_{1:T}, \mathbf{s}_{1:T})\right] \\
&\quad + D_{\mathrm{KL}}(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}))
\end{aligned}
$$

Since $D_{\mathrm{KL}}\left(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T})\right) \geq 0$, we obtain the following evidence lower bound:

$$
\begin{aligned}
\log p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) \geq &\mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}|\mathbf{h}_{1:T}, \mathbf{s}_{1:T})\right] \\
&- D_{\mathrm{KL}}(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}))
\end{aligned} \tag{7}
$$

Let us now calculate the two terms of this lower bound separately. On the one hand:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}|\mathbf{h}_{1:T}, \mathbf{s}_{1:T})\right] \\
&= \mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log \prod_{t=1}^{T} p(\mathbf{o}_t|\mathbf{h}_t, \mathbf{s}_t) p(\mathbf{y}_t|\mathbf{h}_t, \mathbf{s}_t) p(\mathbf{a}_t|\mathbf{h}_t, \mathbf{s}_t)\right] &(8) \\
&= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}_{1:t}, \mathbf{s}_{1:t} \sim q(\mathbf{h}_{1:t}, \mathbf{s}_{1:t}|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}\left[\log p(\mathbf{o}_t|\mathbf{h}_t, \mathbf{s}_t) + \log p(\mathbf{y}_t|\mathbf{h}_t, \mathbf{s}_t) + \log p(\mathbf{a}_t|\mathbf{h}_t, \mathbf{s}_t)\right] &(9)
\end{aligned}
$$

where Equation (8) follows from Equation (2), and Equation (9) was obtained by integrating over remaining latent variables $(\mathbf{h}_{t+1:T}, \mathbf{s}_{t+1:T})$.

On the other hand:

$$
\begin{aligned}
&D_{\mathrm{KL}}(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T})) \\
&= \mathbb{E}_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T} \sim q_{H,S}}\left[\log \frac{q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1})}{p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T})}\right] \\
&= \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \log \frac{q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1})}{p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T})} \,\mathrm{d}\mathbf{h}_{1:T} \,\mathrm{d}\mathbf{s}_{1:T} \\
&= \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \log \left[\prod_{t=1}^{T} \frac{q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})}\right] \mathrm{d}\mathbf{h}_{1:T} \,\mathrm{d}\mathbf{s}_{1:T} &(10) \\
&= \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \log \left[\prod_{t=1}^{T} \frac{q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})}\right] \mathrm{d}\mathbf{h}_{1:T} \,\mathrm{d}\mathbf{s}_{1:T} &(11) \\
& &(12)
\end{aligned}
$$

where:

- Equation (10) follows from the factorisations defined in Equation (2) and Equation (5).
- The simplification in Equation (11) results of $q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) = p(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})$.

Thus:

$$
\begin{aligned}
&D_{\mathrm{KL}}(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T})) \\
&= \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} \left(\prod_{t=1}^{T} q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \right) \left(\sum_{t=1}^{T} \log \frac{q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})} \right) \mathrm{d}\mathbf{h}_{1:T} \, \mathrm{d}\mathbf{s}_{1:T} \\
&= \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} \left(\prod_{t=1}^{T} q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \right) \left(\log \frac{q(\mathbf{s}_1|\mathbf{o}_1)}{p(\mathbf{s}_1)} \right. \\
&\qquad\qquad \left. + \sum_{t=2}^{T} \log \frac{q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})} \right) \mathrm{d}\mathbf{h}_{1:T} \, \mathrm{d}\mathbf{s}_{1:T} \\
&= \mathbb{E}_{\mathbf{s}_1 \sim q(\mathbf{s}_1|\mathbf{o}_1)} \left[\log \frac{q(\mathbf{s}_1|\mathbf{o}_1)}{p(\mathbf{s}_1)} \right] \\
&\quad + \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} \left(\prod_{t=1}^{T} q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \right) \left(\sum_{t=2}^{T} \log \frac{q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})} \right) \mathrm{d}\mathbf{h}_{1:T} \, \mathrm{d}\mathbf{s}_{1:T} \qquad (13) \\
&= D_{\mathrm{KL}}(q(\mathbf{s}_1|\mathbf{o}_1) \,||\, p(\mathbf{s}_1)) \\
&\quad + \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} \left(\prod_{t=1}^{T} q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \right) \left(\log \frac{q(\mathbf{s}_2|\mathbf{o}_{1:2}, \mathbf{a}_1)}{p(\mathbf{s}_2|\mathbf{h}_1, \mathbf{s}_1)} \right. \\
&\qquad\qquad \left. + \sum_{t=3}^{T} \log \frac{q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})} \right) \mathrm{d}\mathbf{h}_{1:T} \, \mathrm{d}\mathbf{s}_{1:T} \\
&= D_{\mathrm{KL}}(q(\mathbf{s}_1|\mathbf{o}_1) \,||\, p(\mathbf{s}_1)) + \mathbb{E}_{\mathbf{h}_1, \mathbf{s}_1 \sim q(\mathbf{h}_1, \mathbf{s}_1|\mathbf{o}_1)} \left[D_{\mathrm{KL}}(q(\mathbf{s}_2|\mathbf{o}_{1:2}, \mathbf{a}_1) \,||\, p(\mathbf{s}_2|\mathbf{h}_1, \mathbf{s}_1)) \right] \\
&\quad + \int_{\mathbf{h}_{1:T}, \mathbf{s}_{1:T}} \left(\prod_{t=1}^{T} q(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1}) q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \right) \left(\sum_{t=3}^{T} \log \frac{q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})}{p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})} \right) \mathrm{d}\mathbf{h}_{1:T} \, \mathrm{d}\mathbf{s}_{1:T} \qquad (14)
\end{aligned}
$$

where Equation (13) and Equation (14) were obtained by splitting the integral in two and integrating over remaining latent variables. By recursively applying this process on the sum of logarithms indexed by t, we get:

$$
\begin{aligned}
&D_{\mathrm{KL}}(q(\mathbf{h}_{1:T}, \mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T-1}) \,||\, p(\mathbf{h}_{1:T}, \mathbf{s}_{1:T})) \\
&= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}_{1:t-1}, \mathbf{s}_{1:t-1} \sim q(\mathbf{h}_{1:t-1}, \mathbf{s}_{1:t-1}|\mathbf{o}_{\leq t-1}, \mathbf{a}_{<t-1})} \left[D_{\mathrm{KL}}(q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \,||\, p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})) \right] \qquad (15)
\end{aligned}
$$

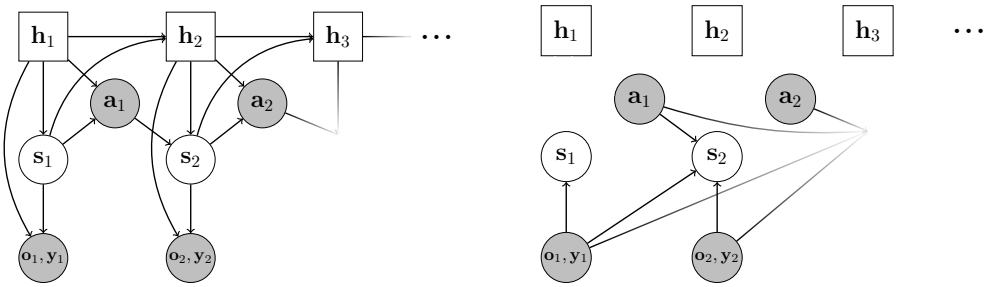
Finally, we inject Equation (9) and Equation (15) in Equation (7) to obtain the desired lower bound:

$$
\begin{aligned}
&\log p(\mathbf{o}_{1:T}, \mathbf{y}_{1:T}, \mathbf{a}_{1:T}) \\
&\geq \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}_{1:t}, \mathbf{s}_{1:t} \sim q(\mathbf{h}_{1:t}, \mathbf{s}_{1:t}|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})} \left[\log p(\mathbf{o}_t|\mathbf{h}_t, \mathbf{s}_t) + \log p(\mathbf{y}_t|\mathbf{h}_t, \mathbf{s}_t) + \log p(\mathbf{a}_t|\mathbf{h}_t, \mathbf{s}_t) \right] \\
&\quad - \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}_{1:t-1}, \mathbf{s}_{1:t-1} \sim q(\mathbf{h}_{1:t-1}, \mathbf{s}_{1:t-1}|\mathbf{o}_{\leq t-1}, \mathbf{a}_{<t-1})} \left[D_{\mathrm{KL}}(q(\mathbf{s}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \,||\, p(\mathbf{s}_t|\mathbf{h}_{t-1}, \mathbf{s}_{t-1})) \right] \square
\end{aligned}
$$

C Model Description

We give a full description of MILE. The graphical models of the generative and inference models are depicted in Figure 8. Table 9 shows the number of parameters of each component of the model, and Table 10 contains all the hyperparameters used during training. Table 11 describes the inference network, and Table 12 the generative network.

C.1 Graphical Models



(a) Generative model. (b) Inference model.

Figure 8: Graphical models representing the conditional dependence between states. Deterministic and stochastic states are represented by, respectively, squares and circles. Observed states are in gray.

C.2 Network Description

Table 9: Parameters of the model.

	Name	Parameters
Inference model ϕ	Observation encoder e_ϕ	34.9M
	Posterior network (μ_ϕ, σ_ϕ)	3.9M
Generative model θ	Prior network $(\mu_\theta, \sigma_\theta)$	2.1M
	Recurrent cell f_θ	6.9M
	BeV decoder l_θ	34.2M
	Policy π_θ	5.9M

C.3 Details on the Network and on Training.

Lifting to 3D. The Lift operation can be detailed as follows: (i) Using the inverse intrinsics K^{-1} and predicted depth, the features in the pixel image space are lifted to 3D in camera coordinates with a pinhole camera model, (ii) the rigid body motion M transforms the 3D camera coordinates to 3D vehicle coordinates (center of inertia of the ego-vehicle).

Observation dropout. At training time the priors are trained to match posteriors through the KL divergence, however they are not necessarily optimised for robust long term future prediction. Hafner et al. [4] optimised states for robust multi-step predictions by iteratively applying the transition model and integrating out intermediate states. In our case, we supervise priors unrolled with random temporal horizons (i.e. predict states at $t + k$ with $k \geq 1$). More precisely, during training, with probability p_{drop} we sample the stochastic state $\mathbf{s}_t$ from the prior instead of the posterior. We call this observation dropout. If we denote X the random variable representing the k number of times a prior is unrolled, X follows a geometric distribution with probability of success $(1 - p_{\text{drop}})$. Observation dropout resembles z-dropout from Henaff et al. [53], where the posterior distribution is modelled as a mixture of two Gaussians, one of which comes from the prior. During training, some posterior variables are randomly dropped out, forcing other posterior variables to maximise their information extraction from input images. Observation dropout can be seen as a global variant of z-dropout since it drops out all posterior variables together.

Table 10: Hyperparameters.

Category	Name	Value
Training	GPUs	8 Tesla V100
	batch size	64
	precision	Mixed precision (16-bit)
	iterations	5×10^4
Optimiser	name	AdamW
	learning rate	1×10^{-4}
	weight decay	1×10^{-2}
	β_1	0.9
	β_2	0.999
	ε	1×10^{-8}
	scheduler	OneCycleLR
	pct start	0.2
Input image	size	600×960
	crop	$[64, 138, 896, 458]$ (left, top, right, bottom)
	field of view	$100°$
	camera position	$[-1.5, 0.0, 2.0]$ (forward, right, up)
	camera rotation	$[0.0, 0.0, 0.0]$ (pitch, yaw, roll)
BeV label	size $H_b \times W_b$	192×192
	resolution b_{res}	0.2m/pixel
Sequence	length T	12
	frequency	5 Hz
	observation dropout p_{drop}	0.25
Loss	action weight	1.0
	image weight	0.0
	segmentation weight	0.1
	segmentation top-k	0.25
	instance weight	0.1
	instance center weight	200.0
	instance offset weight	0.1
	image weight	0.0
	kl weight	1×10^{-3}
	kl balancing	0.75

Additional details The action space is in $\mathbb{R}^2$ with the first component being the acceleration in $[-1, 1]$. Negative values correspond to braking, and positive values to throttle. The second component is steering in $[-1, 1]$, with negative values corresponding to turning left, and positive values to turning right. For simplicity, we have set the weight parameter of the image reconstruction to 0. In order to improve reconstruction of the bird's-eye view vehicles and pedestrians, we also include an instance segmentation loss [54]. Finally, we use the KL balancing technique from [5].

Table 11: Inference model ϕ.

Category	Layer	Output size	Parameters
Image encoder e_ϕ	Input	$3 \times 320 \times 832 = \mathbf{o}_t$	0
	ResNet18 [55]	$[128 \times 40 \times 104, 256 \times 20 \times 52, 512 \times 10 \times 26]$	11.2M
	Feature aggregation	$64 \times 40 \times 104 = \mathbf{u}_t$	0.5M
	Depth	$37 \times 40 \times 104 = \mathbf{d}_t$	0.5M
	Lifting to 3D	$64 \times 37 \times 40 \times 104$	0
	Pooling to BeV	$64 \times 48 \times 48 = \mathbf{b}_t$	0
Route map encoder e_ϕ	Input	$3 \times 64 \times 64 = \mathbf{route}_t$	0
	ResNet18 [55]	$16 = \mathbf{r}_t$	11.2M
Speed encoder e_ϕ	Input	$1 = \mathbf{speed}_t$	0
	Dense layers	$16 = \mathbf{m}_t$	304
Compressing to 1D e_ϕ	Input	$[64 \times 48 \times 48, 16, 16] = [\mathbf{b}_t, \mathbf{r}_t, \mathbf{m}_t]$	0
	ResNet18 [55]	$512 = \mathbf{x}_t$	11.5M
Posterior network (μ_ϕ, σ_ϕ)	Input	$[1024, 512] = [\mathbf{h}_t, \mathbf{x}_t]$	0
	Dense layers	$[512, 512]$	3.9M

Table 12: Generative model θ.

Category	Layer	Output size	Parameters
Prior network $(\mu_\theta, \sigma_\theta)$	Input	$1024 = \mathbf{h}_t$	0
	Dense layers	$[512, 512]$	2.1M
Recurrent cell f_θ	Input	$[1024, 512, 2] = [\mathbf{h}_t, \mathbf{s}_t, \mathbf{a}_t]$	0
	Action layer	64	192
	Pre GRU layer	1024	0.6M
	GRU cell	$1024 = \mathbf{h}_{t+1}$	6.3M
BeV decoder l_θ	Input	$[512 \times 3 \times 3, 1024, 512] = [\text{constant}, \mathbf{h}_t, \mathbf{s}_t]$	0
	Adaptive instance norm	$512 \times 3 \times 3$	1.6M
	Conv. instance norm	$512 \times 3 \times 3$	3.9M
	Upsample conv. instance norm	$512 \times 6 \times 6$	7.9M
	Upsample conv. instance norm	$512 \times 12 \times 12$	7.9M
	Upsample conv. instance norm	$512 \times 24 \times 24$	7.9M
	Upsample conv. instance norm	$256 \times 48 \times 48$	3.3M
	Upsample conv. instance norm	$128 \times 96 \times 96$	1.2M
	Upsample conv. instance norm	$64 \times 192 \times 192$	0.5M
	Output layer	$[8 \times 192 \times 192, 1 \times 192 \times 192, 2 \times 192 \times 192]$	715
Policy π_θ	Input	$[1024, 512] = [\mathbf{h}_t, \mathbf{s}_t]$	0
	Dense layers	2	5.9M

D Experimental Setting

D.1 Dataset

Each run was randomised with a different start and end position, as well as with traffic agents [9]. A random number of vehicles and pedestrians were spawned in the environment as specified in Table 13.

Table 13: Uniform sampling intervals of spawned vehicles and pedestrians in each town during training.

Town	Number of vehicles	Number of pedestrians
Town01	$[80, 160]$	$[80, 160]$
Town03	$[40, 100]$	$[40, 100]$
Town04	$[100, 200]$	$[40, 120]$
Town06	$[80, 160]$	$[40, 120]$

D.2 Metrics

We report metrics from the CARLA challenge [45] to measure on-road performance: route completion, infraction penalty, and driving score.

- **Route completion** $R_{\text{completion}} \in [0, 1]$: for a given simulation scenario, the percentage of route completed by the driving agent. The simulation can end early if the agent deviates from the desired route by more than 30m, or does not take any action for 180s.
- **Infraction penalty** I_{penalty}: multiplicative penalty due to various infractions from the agent (collision with pedestrians/vehicles/static objects, running red lights etc.). $I_{\text{penalty}} \in [0, 1]$, with $I_{\text{penalty}} = 1$ meaning no infraction was observed.
- **Driving score** D: measures both how far the agent drives on the given route, but also how well it drives. D is defined as $D = R_{\text{completion}} \times I_{\text{penalty}} \in [0, 1]$, with $D = 1$ corresponding perfect driving. For a full description of these metrics, please refer to [45].

We now define how the normalised cumulative reward is defined. At every timestep, the environment computes a reward $r \in [R_{\min}, 1]$ [46] for the driving agent. If N is the number of timesteps the agent was deployed for without hitting a termination criteria, then the **cumulative reward** $R \in [N \times R_{\min}, N]$. In order to account for the length of the simulation (due to various stochastic events, it can be longer or shorter), we also report the **normalised cumulative reward** $\overline{R} = R/N$.

We also wanted to highlight the limitations of the driving score as it is obtained by multiplying the route completion with the infraction penalty. The route completion (in $[0, 1]$) can be understood as the recall: how far the agent has travelled along the specified route. The infraction penalty (also in $[0, 1]$) starts at 1.0 and decreases with each infraction with multiplicative penalties. It can be understood as the precision: how many infractions has the agent successfully avoided. Therefore, two models are only comparable at a given recall (or route completion), as the more miles are driven, the more likely the agent risks causing infractions. We instead suggest reporting the cumulative reward in future, that overcomes the limitations of the driving score by being measured at the timestep level. The more route is driven, the more rewards are accumulated along the way. This reward is however modulated by the driving abilities of the model (and can be negative when encountering hard penalties).

D.3 Evaluation Settings

We measure the performance of our model on two settings. Each evaluation is repeated three times.

- **New town, new weathers**: the 10 test scenarios in Town05 [45], on 4 unseen weather conditions: SoftRainSunset, WetSunset, CloudyNoon, MidRainSunset.
- **Train town, train weathers**: the 20 train scenarios in Town03 [45], on 4 train weather conditions: ClearNoon, WetNoon, HardRainNoon, ClearSunset.