DriveWorld: 4D Pre-trained Scene Understanding via World Models for Autonomous Driving

Supplementary Material

6. Pre-training Objective

The proposed DriveWorld for 4D driving pre-training encompasses the following five components:

$$\begin{aligned}
\text{BEV Representation Model}: b_t &\sim q_\phi(b_t \mid o_t) \\
\text{Stochastic State Model}: s_t &\sim q_\phi(s_t \mid h_t, a_{t-1}, o_t) \\
\text{Dynamic Transition Model}: s_t &\sim p_\theta(s_t \mid h_t, \hat{a}_{t-1}) \\
\text{Static Propagation Model}: \hat{b} &\sim p_\theta(\hat{b} \mid b^{'}) \\
\text{Action Decoder}: \hat{a}_t &\sim p_\theta(\hat{a}_t \mid h_t, s_t) \\
\text{3D Occupancy Decoder}: \hat{y}_t &\sim p_\theta(\hat{y}_t \mid h_t, s_t, \hat{b}).
\end{aligned} \tag{4}$$

The joint probability distribution for DriveWorld is:

$$\begin{aligned}
&p(h_{1:T}, s_{1:T}, y_{1:T+L}, a_{1:T+L}) = \\
&\prod_{t=1}^{T} p(h_t, s_t | h_{t-1}, s_{t-1}, a_{t-1}) p(y_t, a_t | h_t, s_t, \hat{b}) \\
&\prod_{k=1}^{L} p(h_k, s_k | h_T, s_T, a_{k-1}) p(y_k, a_k | h_T, s_T, \hat{b}),
\end{aligned} \tag{5}$$

with

$$p(h_t, s_t | h_{t-1}, s_{t-1}, a_{t-1}) = p(h_t | h_{t-1}, s_{t-1}) p(s_t | h_t, a_{t-1}), \tag{6}$$

$$p(y_t, a_t | h_t, s_t) = p(y_t | h_t, s_t, \hat{b}) p(a_t | h_t, s_t), \tag{7}$$

$$p(y_k, a_k | h_T, s_T) = p(y_k | h_T, s_T, \hat{b}) p(a_k | h_T, s_T). \tag{8}$$

Given that h_t is deterministic [22, 27, 97], we have $p(h_t|h_{t-1}, s_{t-1}) = \delta(h_t - f_\theta(\hat{h}_{t-1}, \mathrm{MLN}(\mathrm{s_{t-1}})))$. Consequently, to maximize the marginal likelihood of $p(y_{1:T+L}, a_{1:T+L})$, it is imperative to infer the latent variables $s_{1:T}$. This is achieved through deep variational inference, wherein we introduce a variational distribution $q_{H,S}$ defined and factorized as follows:

$$\begin{aligned}
q_{H,S} &\triangleq q(h_{1:T}, s_{1:T} | o_{1:T}, y_{1:T+L}, a_{1:T+L}) \\
&= \prod_{t=1}^{T} q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t}).
\end{aligned} \tag{9}$$

We parameterise this variational distribution with a neural network with weights ϕ. By formalizing the above process as a generative probabilistic model, we can obtain a variational lower bound on the log evidence:

$$\begin{aligned}
&\log p(y_{1:T+L}, a_{1:T+L}) \geq \mathcal{L}(y_{1:T+L}, a_{1:T+L}; \theta, \phi) \\
&\triangleq \sum_{t=1}^{T} \mathbb{E}_{h_{1:t}, s_{1:t} \backsim q(h_{1:t}, s_{1:t} | o_{\leq t}, a_{<t})} [\underbrace{\log p(y_t | h_t, s_t, \hat{b})}_{\text{past occupancy loss}} \\
&+ \underbrace{\log p(a_t | h_t, s_t)}_{\text{past action loss}}] + \sum_{k=1}^{L} \mathbb{E}_{h_T, s_T \backsim q(h_T, s_T | o_{\leq T}, a_{<T})} \\
&[\underbrace{\log p(y_{T+k} | h_T, s_T, \hat{b})}_{\text{future occupancy loss}} + \underbrace{\log p(a_{T+k} | h_T, s_T)}_{\text{future action loss}}] \\
&- \sum_{t=1}^{T} \mathbb{E}_{h_{1:t-1}, s_{1:t-1} \backsim q(h_{1:t-1}, s_{1:t-1} | o_{\leq t-1}, a_{<t-1})} \\
&[\underbrace{D_{KL}(q(s_t | o_{\leq t}, a_{<t}) \parallel p(s_t | h_{t-1}, s_{t-1}))}_{\text{posterior and prior matching KL loss}}].
\end{aligned} \tag{10}$$

In Eqn. 10, we model $q(s_t|o_{1:t}, a_{1:t-1})$ as a Gaussian distribution, allowing for the closed-form computation of the Kullback-Leibler (KL) divergence. The modelling of actions as a Laplace distribution and 3D occupancy labels as a categorical distribution results in L1 and cross-entropy losses, respectively.

7. Lower Bound Derivation

Next, we will derive the variational lower bound in Eqn. 10. Let $q_{H,S} \triangleq q(h_{1:T}, s_{1:T}|o_{1:T}, y_{1:T+L}, a_{1:T+L})$ be the variational distribution and $p(h_{1:T}, s_{1:T}|a_{1:T+L}, y_{1:T+L})$ be the posterior distribution. The Kullback-Leibler divergence between these two distributions is:

$$\begin{aligned}
&D_{KL}(q(h_{1:T}, s_{1:T}|o_{1:T}, y_{1:T+L}, a_{1:T+L}) \\
&\quad \parallel p(h_{1:T}, s_{1:T}|y_{1:T+L}, a_{1:T+L})) \\
&= \mathbb{E}_{h_{1:T}, s_{1:T} \backsim q_{H,S}} [\log \tfrac{q(h_{1:T}, s_{1:T}|o_{1:T}, y_{1:T+L}, a_{1:T+L})}{p(h_{1:T}, s_{1:T}|y_{1:T+L}, a_{1:T+L})}] \\
&= \mathbb{E}_{h_{1:T}, s_{1:T} \backsim q_{H,S}} \\
&\quad [\log \tfrac{q(h_{1:T}, s_{1:T}|o_{1:T}, y_{1:T+L}, a_{1:T+L}) p(y_{1:T+L}, a_{1:T+L})}{p(h_{1:T}, s_{1:T}) p(y_{1:T+L}, a_{1:T+L}|h_{1:T}, s_{1:T})}] \\
&= \log p(y_{1:T+L}, a_{1:T+L}) - \\
&\quad \mathbb{E}_{h_{1:T}, s_{1:T} \backsim q_{H,S}} [\log p(y_{1:T+L}, a_{1:T+L}|h_{1:T}, s_{1:T})] + \\
&\quad D_{KL}(q(h_{1:T}, s_{1:T}|o_{1:T}, y_{1:T+L}, a_{1:T+L}) \parallel p(h_{1:T}, s_{1:T})).
\end{aligned} \tag{11}$$

Since $D_{KL}(q(h_{1:T}, s_{1:T}|o_{1:T}, y_{1:T+L}, a_{1:T+L}) \parallel p(h_{1:T}, s_{1:T}|y_{1:T+L}, a_{1:T+L})) \geq 0$, we derive the follow-

ing evidence lower bound:

$$\log p(y_{1:T+L}, a_{1:T+L}) \geq \mathbb{E}_{h_{1:T}, s_{1:T} \backsim q_{H,S}}[\log p(y_{1:T+L}, a_{1:T+L} | h_{1:T}, s_{1:T})] - D_{KL}(q(h_{1:T}, s_{1:T} | o_{1:T}, y_{1:T+L}, a_{1:T+L}) \parallel p(h_{1:T}, s_{1:T})). \tag{12}$$

The two terms of this lower bound can be calculated separately. Firstly:

$$\begin{aligned}
&\mathbb{E}_{h_{1:T}, s_{1:T} \backsim q_{H,S}}[\log p(y_{1:T+L}, a_{1:T+L} | h_{1:T}, s_{1:T})] \\
&= \mathbb{E}_{h_{1:T}, h_{1:T} \backsim q_{H,S}}[\log \prod_{t=1}^{T} p(y_t | h_t, s_t, \hat{b}) p(a_t | h_t, s_t) \\
&\quad \prod_{k=1}^{L} p(y_k | h_T, s_T, \hat{b}) p(a_k | h_T, s_T)] \\
&= \sum_{t=1}^{T} \mathbb{E}_{h_{1:t}, s_{1:t} \backsim q(h_{1:t}, s_{1:t} | o_{\leq t}, a_{<t})}[\log p(y_t | h_t, s_t, \hat{b}) p(a_t | h_t, s_t)] \\
&\quad + \sum_{k=1}^{L} \mathbb{E}_{h_T, s_T \backsim q(h_T, s_T | o_{\leq t}, a_{<t})} \\
&\quad [\log p(y_{T+k} | h_T, s_T, \hat{b}) p(a_{T+k} | h_T, s_T)].
\end{aligned} \tag{13}$$

Secondly, with $q(h_t | h_{t-1}, s_{t-1}) = p(h_t | h_{t-1}, s_{t-1})$, we obtain:

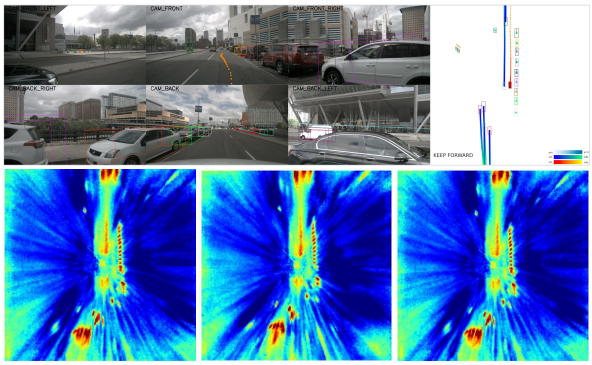
$$\begin{aligned}
&D_{KL}(q(h_{1:T}, s_{1:T} | o_{1:T}, y_{1:T+L}, a_{1:T+L}) \parallel p(h_{1:T}, s_{1:T})) \\
&= D_{KL}(q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1}) \parallel p(h_{1:T}, s_{1:T})) \\
&= \int_{h_{1:T}, s_{1:T}} q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1}) \\
&\quad \log \frac{q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1})}{p(h_{1:T}, s_{1:T})} dh_{1:T} ds_{1:T} \\
&= \int_{h_{1:T}, s_{1:T}} q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1}) \\
&\quad \log [\prod_{t=1}^{T} \frac{q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t})}{p(h_t | h_{t-1}, s_{t-1}) p(s_t | h_{t-1}, s_{t-1})}] dh_{1:T} ds_{1:T} \\
&= \int_{h_{1:T}, s_{1:T}} q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1}) \\
&\quad \log [\prod_{t=1}^{T} \frac{q(s_t | o_{\leq t}, a_{<t})}{p(s_t | h_{t-1}, s_{t-1})}] dh_{1:T} ds_{1:T}.
\end{aligned} \tag{14}$$

Thus:

$$\begin{aligned}
&D_{KL}(q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1}) \parallel p(h_{1:T}, s_{1:T})) \\
&= \int_{h_{1:T}, s_{1:T}} \prod_{t=1}^{T} q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t}) \\
&\quad (\sum_{t=1}^{T} \log \frac{q(s_t | o_{\leq t}, a_{<t})}{p(s_t | h_{t-1}, s_{t-1})}) dh_{1:T} ds_{1:T} \\
&= \int_{h_{1:T}, s_{1:T}} \prod_{t=1}^{T} q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t}) \\
&\quad (\log \frac{q(s_1 | o_1)}{p(s_1)} \\
&\qquad + \sum_{t=2}^{T} \log \frac{q(s_t | o_{\leq t}, a_{<t})}{p(s_t | h_{t-1}, s_{t-1})}) dh_{1:T} ds_{1:T} \\
&= E_{s_1 \backsim q(s_1 | 0_1)}[\log \frac{q(s_1 | o_1)}{p(s_1)}] \\
&\quad + \int_{h_{1:T}, s_{1:T}} (\prod_{t=1}^{T} q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t})) \\
&\quad (\sum_{t=2}^{T} \log \frac{q(s_t | o_{\leq t}, a_{<t})}{p(s_t | h_{t-1}, s_{t-1})}) dh_{1:T} ds_{1:T} \\
&= D_{KL}(q(s_1 | o_1) \parallel p(s_1)) \\
&\quad + \int_{h_{1:T}, s_{1:T}} (\prod_{t=1}^{T} q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t})) \\
&\quad (\log \frac{q(s_2 | o_{1:2}, a_1)}{p(s_2 | h_1, s_1)} \\
&\qquad + \sum_{t=3}^{T} \log \frac{q(s_t | o_{\leq t}, a_{<t})}{p(s_t | h_{t-1}, s_{t-1})}) dh_{1:T} ds_{1:T} \\
&= D_{KL}(q(s_1 | o_1) \parallel p(s_1)) \\
&\quad + \mathbb{E}_{h_1, s_1 \backsim q(h_1, s_1 | o_1}[D_{KL}(q(s_q | o_{1:2}, a_1) \parallel p(s_2 | h_1, s_1))] \\
&\quad + \int_{h_{1:T}, s_{1:T}} (\prod_{t=1}^{T} q(h_t | h_{t-1}, s_{t-1}) q(s_t | o_{\leq t}, a_{<t})) \\
&\quad (\sum_{t=3}^{T} \log \frac{q(s_t | o_{\leq t}, a_{<t})}{p(s_t | h_{t-1}, s_{t-1})}) dh_{1:T} ds_{1:T}.
\end{aligned} \tag{15}$$

Through recursive application of this process to the sum of logarithms indexed by t, we obtain:

$$\begin{aligned}
&D_{KL}(q(h_{1:T}, s_{1:T} | o_{1:T}, a_{1:T-1}) \parallel p(h_{1:T}, s_{1:T})) \\
&= \sum_{t=1}^{T} \mathbb{E}_{h_{1:t-1}, s_{1:t-1} \backsim q(h_{1:t-1}, s_{1:t-1} | o_{\leq t-1}, a_{<t-1})} \\
&\quad [D_{KL}(q(s_t | o_{\leq t}, a_{<t}) \parallel p(s_t | h_{t-1}, s_{t-1}))].
\end{aligned} \tag{16}$$



(a) BEV Features before Prompt (b) Prompt with Mapping (c) Prompt with Forecasting

Figure 5. Visualization of BEV feature maps when prompting with different tasks.

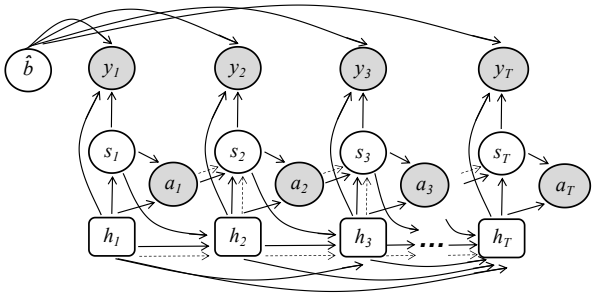


Figure 6. Graphical model of Memory State-Space Model. Deterministic states are denoted by squares, while stochastic states are represented by circles. The observed states are highlighted in grey for clarity. Solid lines represent the generative model, while dotted lines depict variational inference.

Finally, we achieve the intended lower bound:

$$
\begin{aligned}
&\log p(y_{1:T+L}, a_{1:T+L}) \\
&\geq \sum_{t=1}^{T} \mathbb{E}_{h_{1:t}, s_{1:t} \backsim q(h_{1:t}, s_{1:t} | o \leq t, a < t)} [\log p(y_t | h_t, s_t, \hat{b}) + p(a_t | h_t, s_t)] \\
&\quad + \sum_{k=1}^{L} \mathbb{E}_{h_T, s_T \backsim q(h_T, s_T | o \leq T, a < T)} [\log p(y_{T+k} | h_T, s_T, \hat{b}) \\
&\quad + p(a_{T+k} | h_T, s_T)] \\
&\quad - \sum_{t=1}^{T} \mathbb{E}_{h_{1:t-1}, s_{1:t-1} \backsim q(h_{1:t-1}, s_{1:t-1} | o_{\leq t-1}, a_{<t-1})} \\
&\quad [D_{KL}(q(s_t | o_{\leq t}, a_{<t}) \parallel p(s_t | h_{t-1}, s_{t-1}))].
\end{aligned} \tag{17}
$$

8. Dataset

The nuScenes dataset [5] is a large-scale autonomous driving dataset that consists of 700, 150, and 150 sequences for training, validation, and testing, respectively. The scenes are recorded in Boston and Singapore, encompassing a diverse array of weather and lighting conditions, as well as various traffic scenarios.

The OpenScene dataset [11] is the largest 3D occupancy dataset, covering a wide span of over 120 hours of occupancy labels collected in various cities, from Boston, Pittsburgh, Las Vegas to Singapore. OpenScene provides a semantic label for each foreground grid and incorporates the motion information of occupancy flow that helps bridge the gap between decision-making and scene representation. we utilize both semantic occupancy labels and occupancy flow for the supervision of 4D pre-training.

The dense 3D occupancy ground truth is derived by fusing multiple frames of LiDAR point clouds [57, 69]. This approach offers a more comprehensive representation of objects, encompassing details about occluded areas, in contrast to single-frame point clouds. In the future, it may become feasible to directly reconstruct 3D occupancy ground truth from autonomous driving videos using techniques such as NeRF [2, 55], 3D Gaussian Splatting [36], and MVS [80, 81, 89].

9. Task Prompt

During fine-tuning, we add task prompts to BEV maps before each downstream task's decoder. For 3D object detection, the task prompt is "The task is for 3D object detection of the current scene." For planning, the task prompt is "The task involves planning with consideration for both the current and future scenes." The encoder network of task prompts is transferred to downstream tasks, and fine-tuning includes downstream task prompts. This enables different downstream tasks with semantic connections to decouple task-aware features. While basic embeddings for specific tasks are optional, large language model captures complex semantic relationships, providing a nuanced representation of task prompts. Additionally, the strong generalization abilities of such models enhance performance across a wide array of tasks when needed. However, it's worth noting that the current Task Prompt design is relatively simple, and the task number for autonomous driving is limited.

In Fig. 5, we present visualizations of BEV feature maps both before and after the integration of various task prompts. Notably, as shown in Fig. 5 (a), the BEV feature map based on 4D pre-training captures abundant information from both the current and future scenes. While, for specific downstream tasks, some information could be redundant or even detrimental. We utilize task prompts to alleviate the effect of redundant information. In online mapping tasks, the feature map, guided by the task prompt, emphasizes the current spatio-aware information. The targets has more accurate location information in feature map to achieve higher precision . For motion forecasting task, the feature map, guided by the task prompt, conserves both spa-

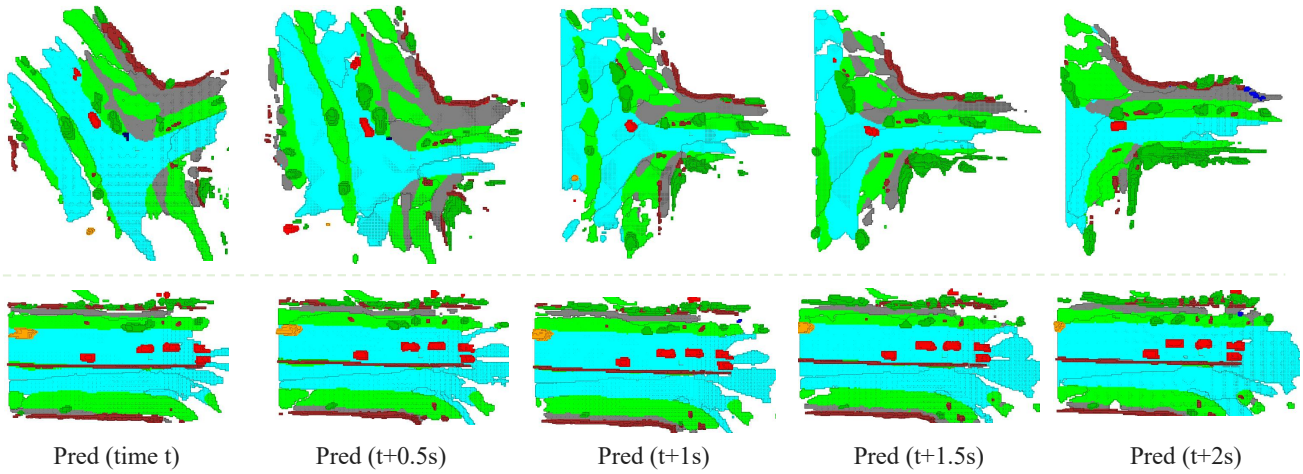


Figure 7. Qualitative example of 3D occupancy predictions, for 2 seconds in the future.

tial and temporal information. The targets cover a broader region in feature map to achieve more robust prediction.

10. Differences between RSSM and MSSM

In world model-based methods such as Dreamers [20, 22, 23] and MILE [27], the RSSM [21] is commonly employed to learn latent variables. However, RSSM, relying on RNN networks, may encounter challenges related to long-term information retention. In contrast, our designed Dynamics Memory Bank in MSSM excels in modelling and preserving long-term information. RSSM compresses features into 1D tensor, while MSSM utilizes context BEV features to reconstruct 3D scenes. Besides, MSSM separates dynamic and static information, addressing them independently.

11. Graphical Model

In Fig. 6, we illustrate the graphical model of the proposed Memory State-Space Model. The update of the deterministic state h_t is dependent on the historical states in Dynamics Memory Bank, facilitating the transmission of temporal-aware features. Spatial-aware features are preserved through the retention of BEV feature $\hat{b}$.

12. Qualitative Results

Fig. 7 presents the reconstruction of both the current and future 3D scenes. This visual representation effectively illustrates DriveWorld's capacity for reconstructing the 3D scene and predicting future changes, thus enhancing downstream task performance after 4D pre-training.