

## 1 Supplementary Material Overview

2 This supplementary material provides additional implementation details, experimental setups, and  
3 extended results to complement the main paper.

- 4 • Section A describes architecture and training details of our image tokenizer and policy  
5 world model, including both pre-training and fine-tuning stages.
- 6 • Section B presents additional experiments on the nuScenes and Closed-loop datasets.
- 7 • Section C provides visualizations of the latent predictions and qualitative comparisons.

## 8 A Implementation and Experimental Details

### 9 A.1 Image Tokenizer

10 **Architecture details.** We adopt a dual-branch architecture to map input images of different resolutions  
11 into a shared latent space. The trainable branch, denoted as  $Q_l$ , is initialized with the same structure  
12 as the frozen high-resolution branch  $Q_h$ . To enhance its modeling capacity, we insert additional  
13 self-attention layers after the multi-level downsampling stages in both the encoder and decoder. These  
14 layers are designed to better capture spatial relationships within the feature maps. Furthermore,  
15 we incorporate cross-attention layers to enable  $Q_h$  to guide the learning of  $Q_l$ . This alleviates the  
16 burden on  $Q_l$  to extract contextual information, allowing it to focus more on modeling temporal  
17 variations and dynamic changes. Specifically, multi-scale features from  $Q_l$  are used as queries, while  
18 the corresponding features from  $Q_h$  serve as keys and values. The attention is applied independently  
19 across scales. In the encoder of  $Q_l$ , self-attention and cross-attention is applied at resolutions of  
20  $8 \times 14$ ; in the decoder of  $Q_l$ , it is applied at  $8 \times 14$ ,  $16 \times 28$ . Additionally, we introduce a lightweight  
21 MLP layer to perform  $4 \times$  downsampling and upsampling of latent token sequence length from  $Q_l$   
22 in the encoder and decoder, respectively. This serves to further compress and reconstruct the latent  
23 representations efficiently.

24 **More training details.** We sample the original Open-Youtube dataset into non-overlapping clips,  
25 each containing 40 frames spanning 4 seconds. During training, each sample randomly selects a  
26 continuous 30-frame segment, from which 2 frames are randomly chosen as high-resolution context  
27 frames. Subsequently, 8 future frames are randomly sampled as low-resolution video inputs. We  
28 optimize the model using the AdamW optimizer with 500 warm-up steps. Image reconstruction is  
29 supervised using the L1 loss. Since pixel-wise differences between future frames and initial context  
30 frames tend to increase over time, we apply a time-dependent weighting to the reconstruction loss,  
31 assigning greater emphasis to later frames to reflect their higher prediction difficulty and encourage  
32 better long-term modeling. We assign a weight of 2.0 to the perceptual loss and a weight of 1.0 to the  
33 discriminator loss to balance perceptual quality and realism during optimization.

### 34 A.2 Policy World Model

#### 35 A.2.1 Pre-training setup

36 Although we sample two consecutive high-resolution initial frames for the tokenizer, only one high-  
37 resolution frame is actually fed into the model per second without supervision. This strategy does not  
38 significantly affect the model performance, while effectively reducing the input token sequence length  
39 and improving training efficiency. Each video clip is randomly cropped into a 24-frame continuous  
40 segment from the original 40-frame clip to define the prediction horizon. As shown in Figure 1(a),  
41 we mark the beginning and end of each high-resolution frame sequence using two special tokens,  
42 " $\langle \text{lsol} \rangle$ " and " $\langle \text{leol} \rangle$ ", following the Show-o convention. For the compressed low-resolution frames,  
43 we introduce two additional special tokens, " $\langle \text{lsodl} \rangle$ " and " $\langle \text{leodl} \rangle$ ", to indicate the start and end  
44 positions of the low-resolution frames.

45 During training, the video autoregressive prediction task serves as the primary objective, while  
46 image-text captioning and pure language modeling tasks are incorporated to preserve the model's  
47 capability in understanding and generating language. These three tasks are mixed in a batch ratio of  
48 3:1:1, respectively. The loss weights are set to 1.0 for the video task and 0.5 for both the captioning  
49 and text-only tasks. The warm-up steps are set to 1000.

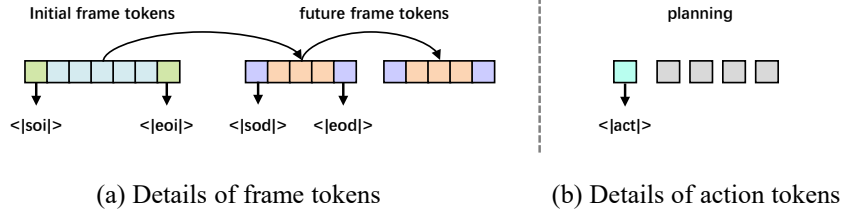


Figure 1: **Detailed structure of input sequences.** (a) illustrates the format of high-resolution and low-resolution video frames. (b) depicts the input configuration used for trajectory prediction.

## 50 A.2.2 Fine-tuning setup.

51 **Details on nuScenes.** We consider two configurations to investigate the influence of ego status on  
 52 planning performance. Structurally, when ego status is included as input, we use a two-layer MLP  
 53 with SiLU activation to project the ego state into the model’s latent space. For navigation commands,  
 54 we randomly initialize three learnable embeddings to represent "Go Straight," "Turn Left," and "Turn  
 55 Right." The "<lactl>" token is used to prompt trajectory prediction, as illustrated in Figure 1(b).  
 56 The action head is implemented as an MLP with SiLU activation. Our framework is designed to  
 57 seamlessly support multi-modal outputs, including both video and language. Given that nuScenes is  
 58 one of the most widely used open-loop datasets and that several prior works have proposed textual  
 59 annotations based on it, we adopt a simplified setting where the action description serves as the target  
 60 for language generation, using fixed prompts. For example: *"In this quiet nighttime driving situation,*  
 61 *the vehicle should maintain a moderate speed and continue straight, adhering to lane discipline."*  
 62 This setup demonstrates the flexibility of our framework and lays the groundwork for future research  
 63 to explore more expressive and diverse multi-modal outputs.

64 During training and evaluation, the last one second of each scene lacks future frames, which makes it  
 65 unsuitable for future frame prediction. Therefore, in the training set, we manually exclude these final  
 66 segments to ensure supervision consistency. For the validation set, we also discard the last one second  
 67 of each scene when evaluating video generation metrics. However, to ensure a fair comparison with  
 68 previous works on planning, we retain these segments when computing planning-related metrics.

69 **Details on NAVSIM.** We follow the official practice by concatenating the ego status and navigation  
 70 command into a single vector, which is then projected into the model space via an MLP layer. Since  
 71 this dataset does not provide textual descriptions, we do not include any language modeling tasks  
 72 during training or inference. Additionally, some video frames required for prediction are not included  
 73 in the NAVSIM dataset. To address this, we resample the missing frames from the original nuPlan  
 74 dataset. Other aspects of the training procedure remain largely consistent with that of nuScenes.

## 75 B More experiments

### 76 B.1 Set up in Dynamic Focal Loss

77 To effectively improve the video modeling and generation capability of the Policy World Model, we  
 78 propose a Dynamic Focal Loss (DFL) that emphasizes temporally varying image regions through  
 79 spatial weighting. We conduct an ablation study on the key hyperparameters  $\alpha$  and  $\beta$  on nuScenes,  
 80 where  $\alpha$  controls the weight for spatial tokens that change over time, and  $\beta$  controls the weight for  
 81 those that remain unchanged across consecutive frames. As shown in Table 1, we fix  $\alpha$  to 1.0 and  
 82 vary  $\beta$  to explore their relative influence.

83 We observe that when  $\alpha < \beta$  and  $\beta$  is set to a small value (e.g.,  $\beta = 0.1$ ), the large disparity in task  
 84 weights leads to overfitting in the future frame prediction task during training, while the planning  
 85 task has not yet fully converged. This imbalance ultimately results in degraded overall performance,  
 86 indicating the need to better coordinate the training progress of the two tasks for more effective joint  
 87 optimization. On the other hand, when  $\alpha \geq \beta$ , the Dynamic Focal Loss mechanism tends to fail,  
 88 leading to a significant drop in the quality of future frame prediction, which in turn negatively impacts  
 89 the performance of the downstream planning task.

Table 1: **Ablation study on the hyperparameters in the Dynamic Focal Loss.** The dynamic weight  $\alpha$  is fixed to 1.0 across all experimental settings.

$\beta$	Visual Forecast Quality			Planning Metrics	
	LPIPS↓	PSNR↑	FVD↓	Avg.L2(m)↓	Avg.Col(%)↓
0.1	0.23	22.69	65.07	0.82	0.12
0.4	0.22	23.07	67.13	0.78	0.07
0.7	0.22	23.11	71.84	0.84	0.09
1.0	0.22	22.88	96.99	1.04	0.26
2.0	0.22	22.65	93.83	1.27	0.27



Figure 2: **Visualization.** Comparison of future frame predictions with and without Dynamic Focal Loss (DFL). The first row shows ground truth frames, the second row shows predictions without DFL, and the third row shows predictions with DFL. Sampled frames at  $t=2, 4, 6, 8, 10$  are shown.

To further illustrate the qualitative impact of DFL, we provide a visual comparison in Figure 2. Each row corresponds to a specific model configuration: the first row shows ground truth future frames, the second row presents prediction results without DFL (with  $\alpha = \beta = 1.0$ , making the loss ineffective), and the third row shows results with DFL enabled (with  $\alpha = 0.1$ ,  $\beta = 0.4$ ). These visualizations demonstrate that the use of DFL helps the model better capture and represent dynamic scene elements over time, resulting in more accurate and temporally coherent predictions.

## B.2 Visualization and Analysis of Temporal Representations in Predicted Video Frames.

As shown in Figure 3, we visualize the 2D UMAP projection of forecasted latent video frames over time on the nuScenes validation split. Specifically, we extract future predicted driving latents from each 20-second-long scene and concatenate all samples across scenes. Each predicted frame is then projected into a 2D coordinate point according to its temporal order. Different colors are used to indicate the temporal progression from 0s to 20s.

We observe that, although the future representations are generated independently for each sample and conditioned on different input frames, the resulting projected embeddings consistently exhibit smooth and coherent temporal dynamics across the full prediction horizon. This phenomenon suggests

Table 2: **Impact of the Textual Generation Task on nuScenes validation split.** We also conduct an ablation study to evaluate the impact of the textual generation task on planning performance. Specifically, we compare different settings: (1) "None": without incorporating any text generation task, (2) "Scene": with scene description prediction, and (3) "Action": with action description prediction.

Text Task Type	Visual Forecast Quality			Planning Metrics	
	LPIPS↓	PSNR↑	FVD↓	Avg.L2(m)↓	Avg.Col(%)↓
None	0.22	23.12	65.45	0.77	0.09
Scene	0.22	22.97	67.52	0.77	0.08
Action	0.22	23.07	67.13	0.78	0.07

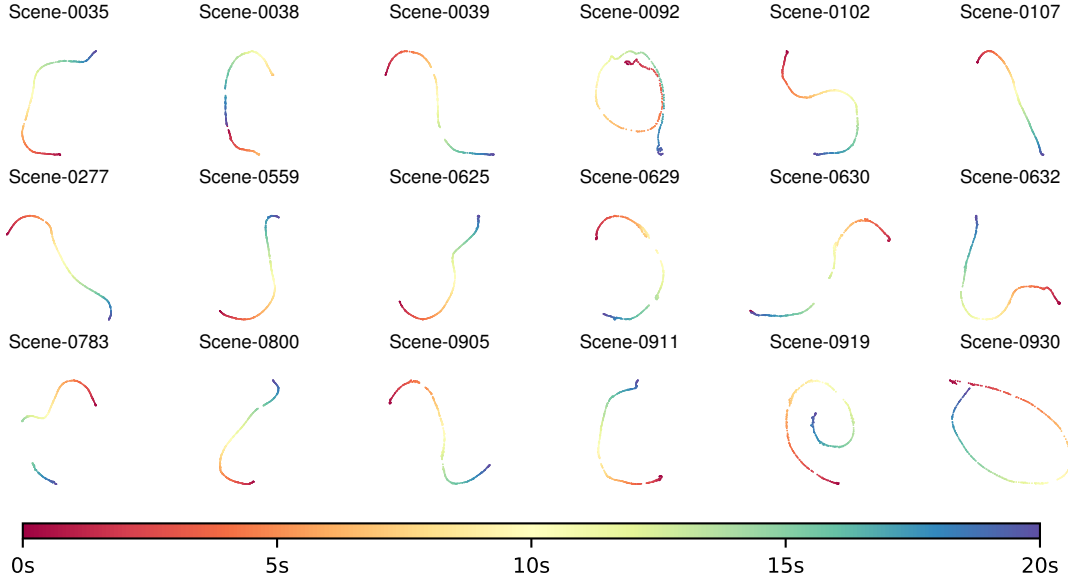


Figure 3: The UMAP projection reveals temporal changes in frame-level latent representations, highlighting the model’s ability to capture scene dynamics across different environments.

that our Policy World Model is capable of learning a robust internal representation of the temporal evolution of driving scenes. Importantly, it also demonstrates that the model can effectively decouple the dynamics from specific visual content in the input, capturing underlying motion patterns that are consistent and semantically meaningful, regardless of the particular observation used as guidance.

### B.3 Impact of the Textual Generation Task on Planning Performance.

To isolate the impact of different types of text prediction on planning performance of nuScenes, we design three settings. The first setting excludes any text prediction task, which also serves as the baseline in the NAVSIM dataset. The second setting provides a detailed description of the scene environment, while the third offers a brief prediction of the ego vehicle’s future behavior. For action descriptions, we filter out information related to multi-view perspectives and retain only a brief summary for supervision. As shown in Table 2, In our model, incorporating scene- or action-level textual generation tasks does not lead to significant improvements in future frame forecasting or downstream planning metrics, suggesting a limited effect in our specific setting.

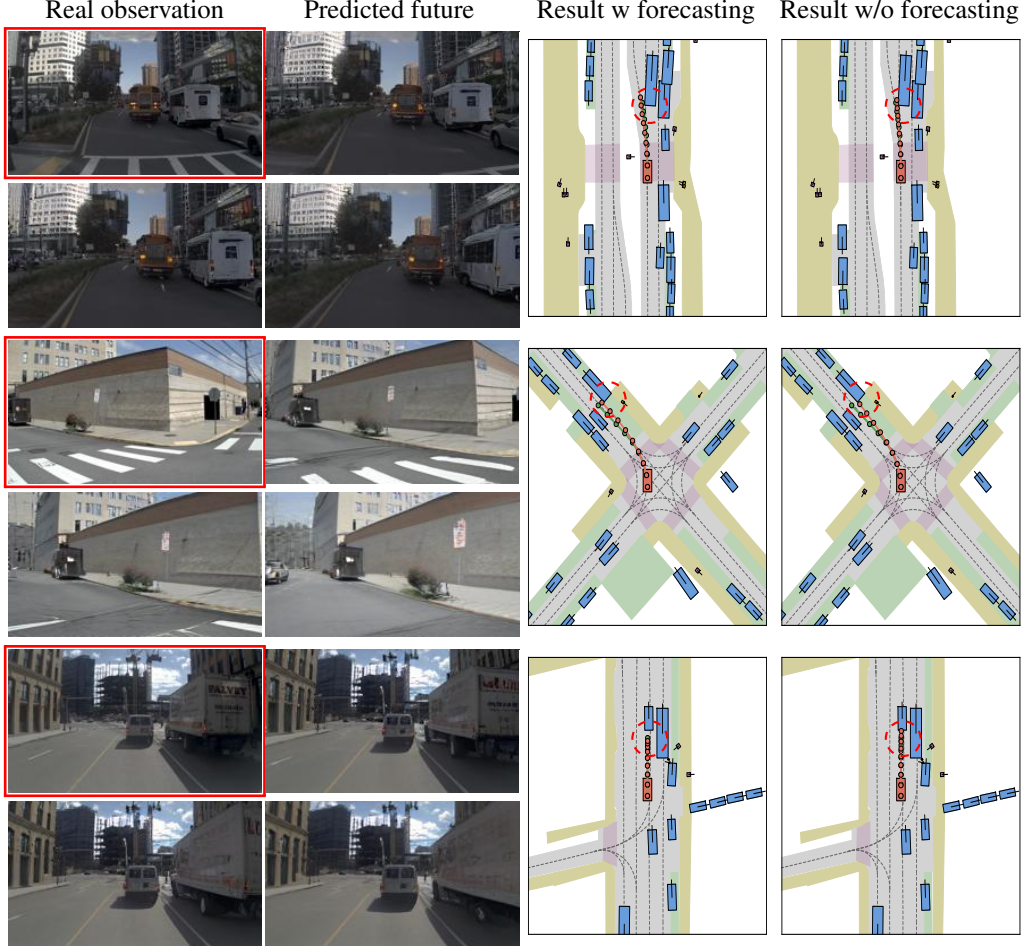


Figure 4: More comparison of planning results with and without incorporating future prediction during training (green: GT, orange: prediction).

## C Visualizations

### C.1 Comparison visualization

In Figures 4 and 5, we provide additional qualitative results to compare planning with and without future frame prediction. On the left, we show a sequence of video frames where the red-bordered frame indicates the current observation, and the unframed ones are predicted future frames. We uniformly sample three future frames for visualization. On the right, we present the corresponding BEV (bird’s-eye view) planning outcomes. These comparisons highlight how incorporating future frame prediction can enhance planning quality by enabling better anticipation of dynamic scene changes. In Figure 6, we show additional visual comparisons illustrating the effect of Dynamic Focal Loss on future frame prediction.





Figure 5: **Visualization.** More comparison of planning results with and without incorporating future prediction during training (green: GT, orange: prediction).

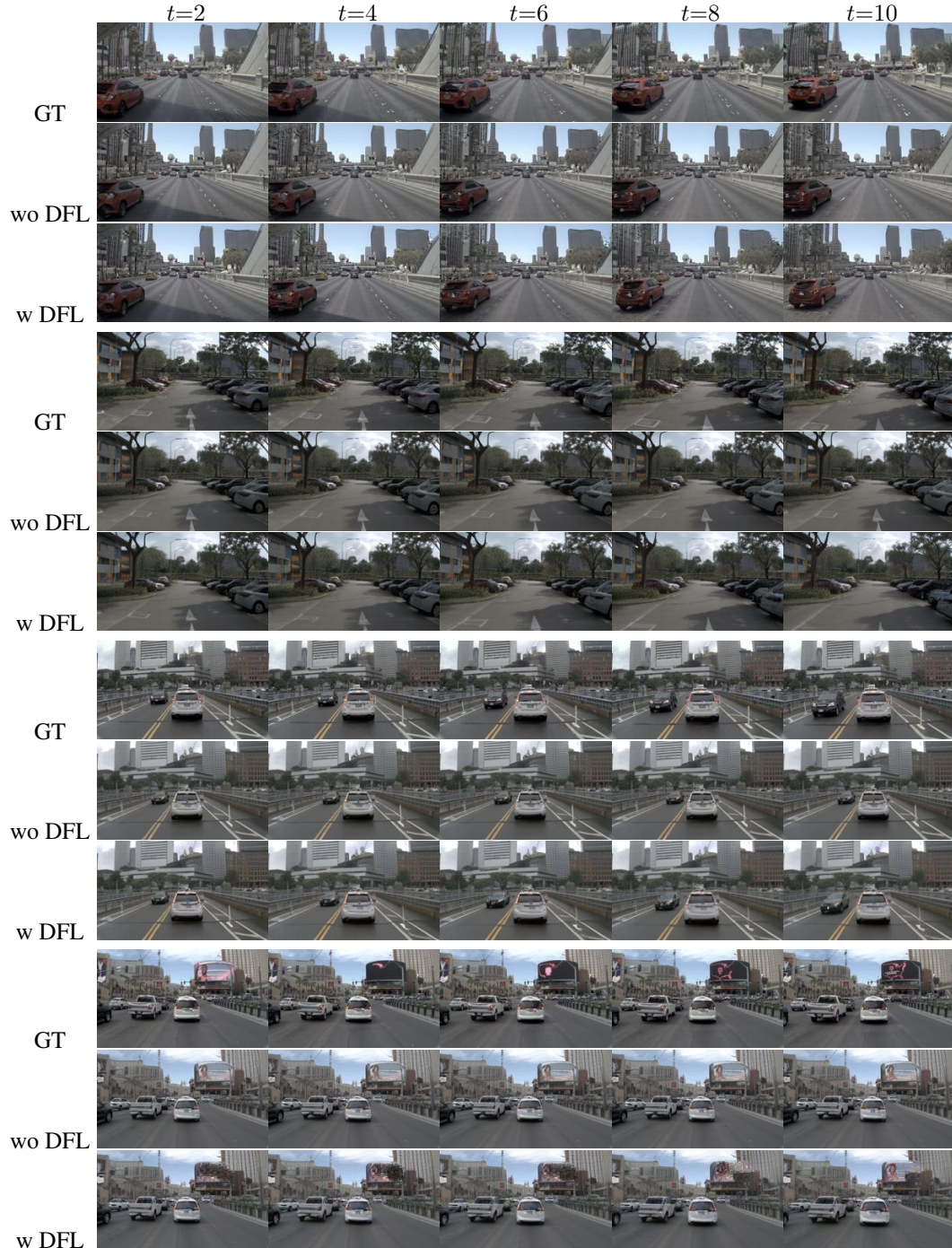


Figure 6: **Visualization.** More comparison of future frame predictions with and without Dynamic Focal Loss (DFL). The first row shows ground truth frames, the second row shows predictions without DFL, and the third row shows predictions with DFL. Sampled frames at  $t=2, 4, 6, 8, 10$  are shown.