Appendix

***Abstract*— This appendix provides additional implementation details and architectural insights to complement the main paper.**

I. Network Framework

Fig. 1 illustrates a more detailed view of the proposed network architecture. The overall framework is composed of several core modules, including:

- **CBR**: Convolution → BatchNorm → ReLU
- **CB**: Convolution → BatchNorm
- **FC**: Fully Connected layer
- **Deconvolution layers** for upsampling
- **1×1 Convolution layers** for dimension adjustment
- **Sigmoid activations** for binary prediction tasks

The encoder backbone extracts hierarchical features through multiple convolutional blocks (`c1` to `c5`), followed by top-down feature fusion via deconvolution layers (`p4`, `p5`). The encoded features are subsequently flattened and passed into the diffusion policy module, which iteratively refines trajectory predictions through N denoising steps.

The entire model is supervised using a combination of Mean Squared Error (MSE) Loss and Binary Cross-Entropy Loss, depending on the output branch.

II. Key Tensor Shapes

The main tensors used in the *TopoDiffuser* framework are summarized in Table I. This table provides the shape and brief description of each tensor that plays a critical role in the forward pass of the model. These include the input representation, intermediate encoder outputs, as well as the final predicted trajectories generated by the diffusion policy.

A. Denoising Process Visualization

We visualize intermediate trajectory predictions at different denoising steps to illustrate how the diffusion process refines noisy samples into feasible trajectories, as shown in Fig. 2.

In the early steps, trajectories show large deviations and poor alignment with the road. As denoising progresses, they become smoother and increasingly follow the topometric route. By the final step, the predictions align closely with the road geometry and ground-truth paths.

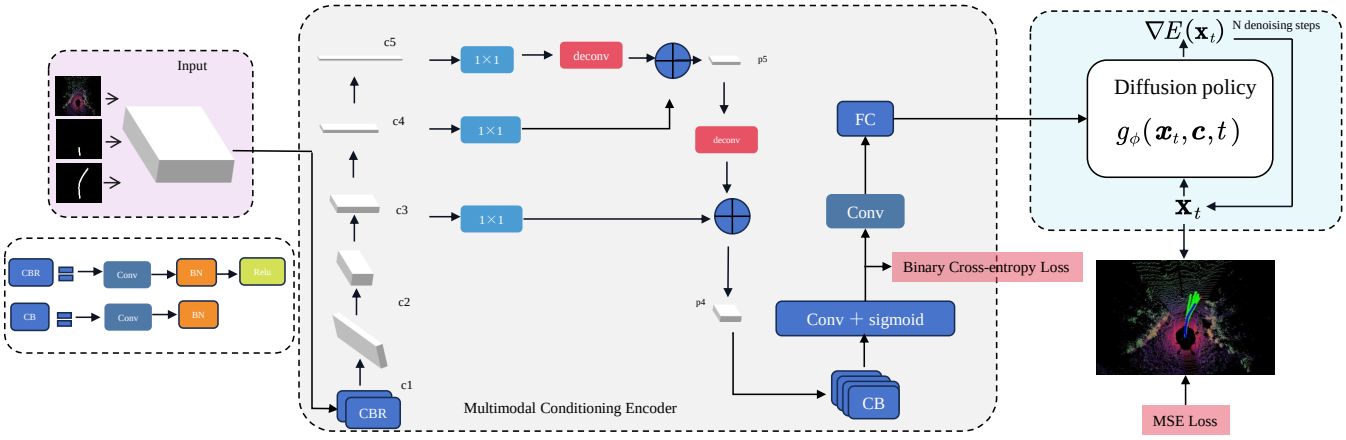

Fig. 1. Overview of the proposed diffusion-based trajectory prediction framework.

TABLE I

KEY TENSOR SHAPES IN *TopoDiffuser*

Tensor Name	Shape	Description
`x_input`	$[8, 5, 300, 400]$	Input tensor: batch size = 8, 5 channels (LiDAR, map, history), BEV image size = 300×400.
`obs_cond (before reshape)`	$[8, 64, 8]$	Intermediate condition features from the encoder.
`obs_cond (after reshape)`	$[8, 512]$	Flattened condition vector (64×8) used as input to the diffusion policy network.
`diffusion_output`	$[40, 8, 2]$	Output trajectories: 5 samples × 8 waypoints = 40 waypoints, each with (x, y) coordinates.

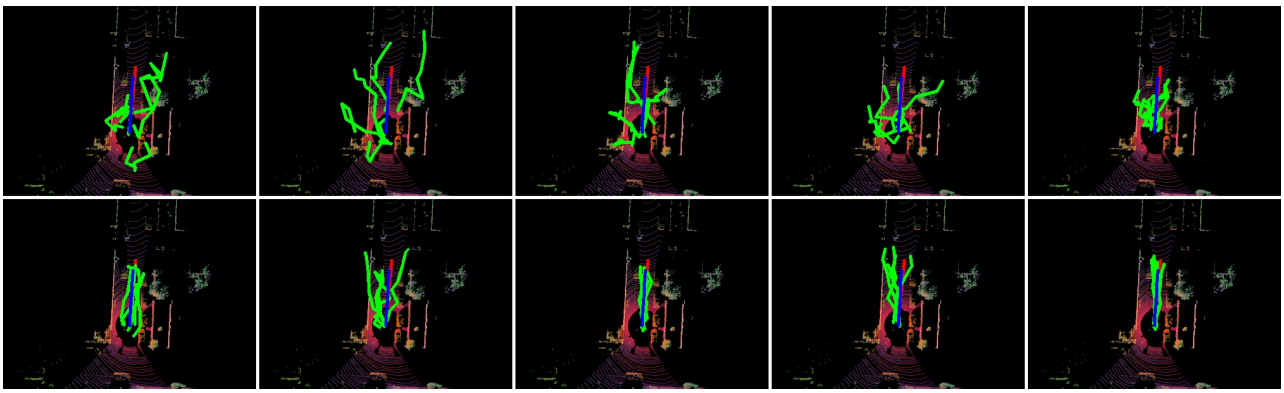

Fig. 2. Trajectory refinement over denoising steps.