Diffusion-FS: Multimodal Free-Space Prediction via Diffusion for Autonomous Driving

Keshav Gupta[1], Tejas S. Stanley[1], Pranjal Paul[1], Arun K. Singh[2] and K. Madhava Krishna[1]

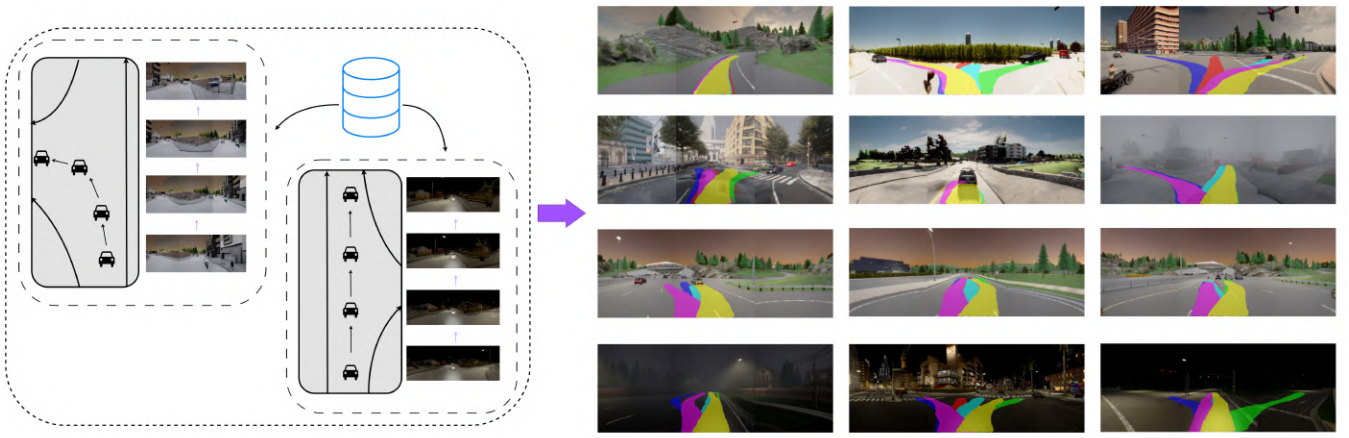

Fig. 1: **Left:** A dataset of raw driving logs containing image and ego trajectory pairs. Our self supervised method processes such an unannotated dataset to generate freespace segments essential for autonomous driving. **Right:** Examples of multimodal freespace segments generated by our diffusion model on CARLA. At inference, our model denoises a fixed number of noise samples into freespace segments. We showcase predictions across various weather conditions, times of day, road topologies, and obstacle layouts.

***Abstract*— Drivable Freespace prediction is a fundamental and crucial problem in autonomous driving. Recent works have addressed the problem by representing the entire non-obstacle road regions as the freespace. In contrast our aim is to estimate the driving corridors that are a navigable subset of the entire road region. Unfortunately, existing corridor estimation methods directly assume a BEV centric representation, which is hard to obtain. In contrast, we frame drivable freespace corridor prediction as a pure image perception task, using only monocular camera input. However such a formulation poses several challenges as one doesn't have the corresponding data for such freespace corridor segments in the image. Consequently, we develop a novel self-supervised approach for freespace sample generation by leveraging future ego trajectories and front-view camera images, making the process of visual corridor estimation dependent on the ego trajectory. We then employ a diffusion process to model the distribution of such segments in the image. However, the existing binary mask based representation for a segment poses many limitations. Therefore, we introduce ContourDiff, a specialized diffusion-based architecture that denoises over contour points rather than relying on binary mask representations, enabling structured and interpretable freespace predictions. We evaluate our approach qualitatively and quantitatively on both NuScenes and CARLA, demonstrating its effectiveness in accurately predicting safe multimodal navigable corridors in the image.**

Project Page - https://keshav0306.github.io/diffusion˙fs/

I. Introduction

The autonomous navigation community is increasingly exploring vision-based approaches that aim to map the observations to actions either through direct perception [1–3] or via intermediate metric scene representation such as occupancy grid or BEV map [4–6]. The former often struggles with vehicle kinematic constraints, obstacle avoidance, and lane boundaries due to reliance on error-prone perception modules that map high-dimensional features to control inputs. In contrast, humans while driving, identify *free-space*[1] rather than enumerating and precisely localizing the obstacles. This free-space perception approach is evident in common driving scenarios. For instance, a driver merging onto a highway focuses on regions or *"corridors"* between vehicles, rather than their exact position. For driving situations that need more informed decisions, the navigable space naturally diversifies into multiple potential paths or corridors, each representing a distinct mode of navigation. For instance, at a T-junction, the driver simultaneously perceives multiple valid navigable options— left turn, right turn, or proceeding straight. Each represents a distinct viable option where the "correct" path depends on contextual factors such as destination intent or traffic flow that are inherently multimodal. Humans instinctively evaluate these multiple plausible paths before committing to one, relying primarily on a relative sense of depth perception regarding surrounding traffic agents, scouting out drivable spaces rather than making precise

[1]Robotics Research Center, IIIT-Hyderabad, India
[2]The University of Tartu, Estonia

[1]The freespace navigation in literature, is understood under two different categories: *1. Drivable-Area Prediction* [7–10], which represents the entire non-obstacle road region, broadly studied as lane segmentation and/or road segmentation task, and *2. Driving-Corridor Estimation* [11–14], which discretize the drivable region into a reachable set where the vehicle can reach over time from its current ego position without collisions. Our interest lies in 2nd, posing it as a perception task.

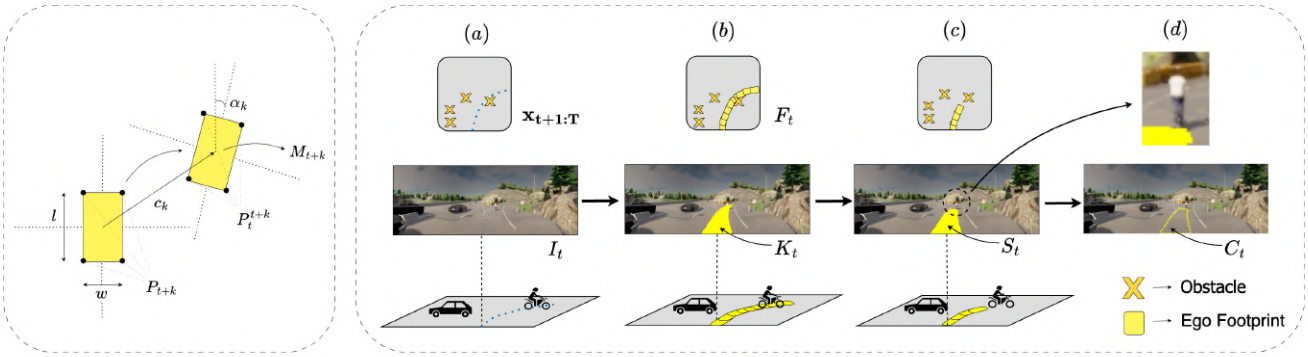

Fig. 3: **Freespace Contour Creation. Left :** We show the transformations between the ego vehicle's frame at time $t+k$ and the frame at time t. **Right** : We show the process of creation of the freespace sample for an image. The top row presents the BEV map in the local frame of the ego vehicle at time t, while the middle and bottom rows show the corresponding frontal camera images and an alternative top-down view, illustrating the camera setup and projection process from a third-person perspective. (a) shows the future ego trajectory of the ego vehicle $x_{t+1:T}$. (b) shows the corresponding future footprint of the ego vehicle, F_t and its projection K_t in the image plane. (c) the future footprint K_t is bounded up to the closest overlapping obstacle to obtain the freespace segment, S_t. (d) the corresponding freespace contour C_t is obtained.

metric calculations.

Applying this intuition to autonomous driving, we explore the prediction of multimodal navigable regions directly from a monocular camera input. While the drivable area prediction task is not entirely novel [7–14], studying them as vision-oriented task remains unexplored. Hence, we pose this problem as visual corridor prediction task which contrasts with existing work [11–14] that assumes prior knowledge of obstacle positions and adopt geometric or optimization-driven strategies to compute navigable regions.

In this paper, we define navigable regions as pixel-level segments in terms of contour points that are a set of collision-free regions in the vicinity of the vehicle, as shown in Figure 6.

Our contributions are the following:

1) We formulate the task of visual corridor prediction as an image perception task for the first time, contrary to prior works that assume the availability of a BEV centric representation.
2) We propose a novel self supervised approach for freespace sample generation from future ego trajectory and images.
3) ContourDiff - We propose a novel diffusion architecture for denoising over contour points rather than a standard binary mask based representation.

II. Related Works

A. Perception-Based Navigable Region Identification

Perception-driven methods for identifying navigable regions in autonomous driving have been widely explored. Works like [15, 16] map linguistic commands to goal regions rather than segmenting navigable space. Freespace segmentation approaches [7–10] classify entire roads as freespace, losing the essence of true navigable regions. Their reliance on supervised learning with labeled datasets limits generalization to diverse road structures. Overcoming these limitations, DiffusionFS adopts a diffusion-based approach to generate multimodal predictions of navigable corridors while maintaining semantic awareness of the surrounding environment. By explicitly avoiding obstacles and off-road areas, it constructs feasible driving corridors from the ego vehicle's perspective.

B. Diffusion Based Segmentation Approaches

Recent advances in diffusion models have enabled their application in segmentation by leveraging generative processes to create or refine segmentation masks. Unlike traditional segmentation approaches, which directly classify pixels, these methods utilize learned diffusion processes to generate structured masks or extract meaningful features from the denoising steps. [17] extracts intermediate activations from pretrained diffusion models, using them as feature representation for segmentation. SegDiff [18] formulates segmentation as a conditional generation problem using Conditional Diffusion Probabilistic Models. However, due to the high dimensionality of segmentation masks, SegDiff suffers from slow inference and fails to capture the data distribution properly. To address this, ContourDiff denoises directly on contour representations rather than full-resolution masks. This significantly reduces dimensionality while preserving essential spatial structure, allowing for efficient segmentation without losing navigable region fidelity.

III. Methodology

We propose a novel self-supervised method for directly detecting freespace in images without requiring any annotated data. Instead, our approach leverages raw driving logs of the ego vehicle, which are naturally abundant, extensive, and easily accessible from large-scale autonomous driving datasets or onboard vehicle sensors. Using a diverse set of pairs of images and corresponding freespaces obtained through this method, we then train a conditional diffusion model over the freespace contours. During inference for an image, one can

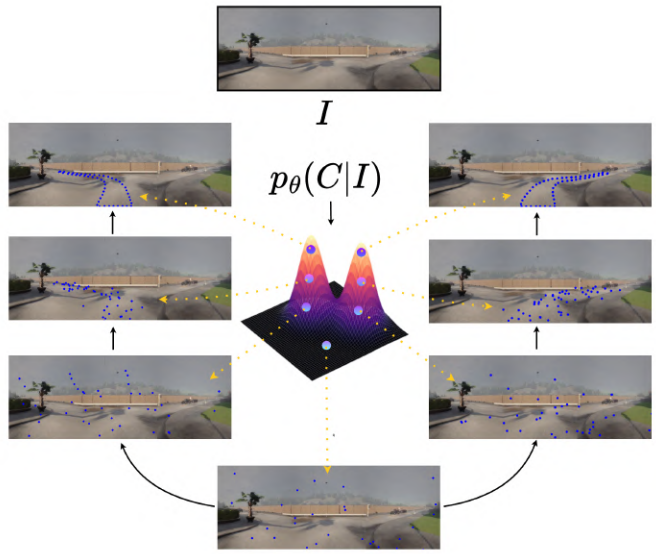

Fig. 4: **Conditional Probability Distribution of Freespace Contours given an image.** We show an example of an intersection where the distribution of freespace contours is likely to be bimodal, as there is possibility of freespace at both the left and the right turn. The training data provides enough evidence to approximate this distribution, as in many driving logs covering a similar scenario, the ego vehicle must have traversed along both ways.

sample multiple freespace segments allowing for scalable, annotation-free, and adaptable freespace prediction.

A. Self Supervised Freespace Generation

Our key observation is that the ego trajectory is inherently correlated with the freespace visible in the image, as we define freespace as the navigable portion of the road that aligns with human driving behavior. Since the ego vehicle always moves within a drivable region, its future positions provide a strong prior for freespace. We define one possible freespace segment by projecting the ego vehicle's future footprints into the image. This segment does not entirely represent freespace, as the ego vehicle might have crossed regions in the future where some obstacle is present for the current timestep. This will yield an overlap of the freespace segment with the obstacle. We assume access to obstacle bounding boxes in the image plane, which can be efficiently obtained using existing object detection models, even for unannotated datasets using [19]. We limit the segment to the closest obstacle to guarantee freespace. As different driving episodes yield varying trajectories for similar scenes, our method captures multiple plausible freespace regions.

A driving log can be represented as a sequence of $\{(I_t, \mathbf{x}_t)\}_{t=1}^{T}$ pairs, where $I_t \in \mathbb{R}^{H\times W\times 3}$ is the image at the current timestep, $\mathbf{x}_t = \{x_t, y_t, \theta_t\} \in \mathbb{R}^3$ is the pose of the ego vehicle, and T is the episode length. For a given timestep t in the driving log, our goal is to determine a possible freespace segment, which depends on the future trajectory of the ego vehicle, $\mathbf{x_{t+1:T}} = \{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_T\}$.

For each future timestep $t+k$, we define the footprint of the ego vehicle, M_{t+k} as the segment corresponding to the rectangular area of the ego vehicle in the local frame of the ego vehicle at timestep t, with the center of the rectangle as

$$c_k = (x_{t+k} - x_t, y_{t+k} - y_t)$$

and the relative orientation as

$$\alpha_k = \theta_{t+k} - \theta_t$$

Formally, M_{t+k} is formed from the 4 points of the oriented bounding box corresponding to the footprint of the ego vehicle. Let P_{t+k} define the set of corner points of the footprint in the frame of the ego vehicle at timestep $t+k$.

$$P_{t+k} = \begin{bmatrix} -\frac{w}{2} & \frac{w}{2} & \frac{w}{2} & -\frac{w}{2} \\ -\frac{l}{2} & -\frac{l}{2} & \frac{l}{2} & \frac{l}{2} \end{bmatrix}$$

where w, l are the width and length of the ego vehicle. We next transform each of these points to the frame of the ego vehicle at timestep t denoted by P_t^{t+k}, by the transformation

$$P_t^{t+k} = R(\alpha_k) \cdot P_{t+k} + c_k$$

where $R(\alpha_k)$ is the 2D rotation matrix corresponding to the angle α_k. The above transformation is visualized in fig. 3. Let p_1, p_2, p_3, p_4 deonte the transformed corner points i.e. columns of P_t^{t+k}. We get the transformed footprint mask M_{t+k} as

$$M_{t+k}(u,v) = \begin{cases} 1, & \text{if } (u,v) \text{ is inside the rectangle formed by } \{p_1,p_2,p_3,p_4\}, \\ 0, & \text{otherwise.} \end{cases}$$

The combined footprint mask, denoted as F_t will be simply the bitwise OR of these masks for all k.

$$F_t = \bigcup_{k=1}^{T-t} M_{t+k}$$

On getting F_t, we project this mask to the camera frame using known camera intrinsics and extrinsic parameters. For every point (u, v) in F_t, we get the projected point in the image frame using the camera intrinsics K, camera rotation matrix R and camera height h using

$$\mathbf{p}' = K \cdot R \cdot \begin{bmatrix} u \\ -h \\ v \end{bmatrix} \tag{1}$$

We denote this transformed mask as K_t. This mask represents the segmented path taken by the ego vehicle in future timesteps as viewed in the image plane in the current timestep, t. Given the set of bounding boxes of all the obstacles in the image plane denoted as O_t, we limit K_t to the nearest obstacle in O_t by finding the closest obstacle overlapping with K_t, from which we get our desired freespace segment S_t. For our experiments, we deal with an equivalent conversion of this segment mask, which is the ordered set of contour points denoted as C_t. We get C_t from S_t using the OpenCV [20] implementation of [21]. Please refer to fig. 3 for a visual illustration of the process.

B. Diffusion Formulation

Predicting the distribution of freespaces in an image can be very challenging because of the complex multimodal nature of the task. To effectively model this, we use a diffusion model [22] to approximate the true conditional distribution $q(C|I)$ of freespace contours C given the image input I through $p_\theta(C|I)$. Figure 4 presents an illustration of the approximated distribution expected through the diffusion process.

Starting from an initial noisy contour $C^{T_{\max}} \sim \mathcal{N}(0, I)$, we iteratively denoise it to obtain a set of contours with decreasing noise levels, $\{C^{T_{\max}}, C^{T_{\max}-1}, \ldots, C^2, C^1, C^0\}$. The denoising conversion from C^t to C^{t-1} follows the equation

$$C^{t-1} = \alpha(C^t - \gamma\epsilon_\theta(C^t, t) + N(0, \sigma^2 I))$$

where α, γ and σ depend on the variance schedule of the diffusion process, and ϵ_θ is the denoising model parameterized by θ.

During training, we have access to a dataset $\mathcal{D}$ consisting of pairs of images and freespace contours, (I_i, C_i). We sample the timestep t uniformly from $(0, T_{max})$, and run the diffusion forward process for t timesteps on the contour, C from a randomly sampled pair (I, C) from our dataset. The diffusion objective is to minimize

$$\min_\theta \mathbb{E}_{\substack{t\sim\mathcal{U}(0,T_{\max}) \\ \epsilon\sim\mathcal{N}(0,I) \\ (I,C)\sim\mathcal{D}}} \left[||\epsilon_\theta(I, C^t, t) - \epsilon||_2^2\right] \tag{2}$$

C. ContourDiff - Denoising Contours

Our proposed diffusion model architecture operates on contour points, offering a more interpretable alternative to diffusion over masks. Unlike mask-based diffusion, our approach ensures that at each timestep of the reverse process, we obtain a set of points directly on the image, providing a clear geometric interpretation of denoising. Such a property eliminates the need for thresholding to binarize the mask, as the point positions inherently define it. Additionally, representing a closed connected mask with points is both natural and efficient, requiring only $N \times 2$ parameters compared to the $H \times W$ parameters needed for a latent mask representation. This structured representation acts as a prior for modeling connected closed segments, making it particularly well-suited for our task. This also improves the convergence during training and output quality. Refer to Figure 5 for details about the architecture.

The input to the model is the set of noisy contour points at the forward process timestep t,

$$C^t = \{x_1^t, x_2^t ... x_N^t\} \in \mathbb{R}^{N\times 2}$$

and the image I. For this, we pass the image through an image encoder F_{enc} first to get the features corresponding to the image.

$$F = F_{enc}(I) \in \mathbb{R}^{H'\times W'\times D_f}$$

where H', W' is the size of the downsampled feature map and D_f is the dimension of the each element of the feature map.

We then extract features at the noisy contour point locations using bilinear sampling, a method that interpolates features at fractional positions within the feature map. For each k-th point in the contour, x_k^t, the bilinearly sampled feature is given by:

$$f_k = \mathrm{B}(F, x_k^t) \in \mathbb{R}^{D_f}$$

where B is the sampling function that takes the feature map F and the sampling location x_k^t as inputs. The features lack any inherent position information, making it essential for the denoising model to be aware of each point's location to accurately estimate the noise. To address this, we concatenate each point's position with its sampled feature. Specifically, we first project the 2D position into a higher-dimensional positional embedding $e_k \in \mathbb{R}^{D_e}$. Thus, for every point, we obtain the feature vector g_k as the concatenation of f_k and e_k, forming a structured set of features corresponding to each contour point in C^t.

$$G = \{g_1, g_2, \ldots, g_N\} \in \mathbb{R}^{N\times D}$$

where

$$g_i = \begin{bmatrix} f_i \\ e_i \end{bmatrix} \in \mathbb{R}^D, \quad \forall i \in \{1, \ldots, N\}, \quad D = D_f + D_e$$

The timestep embedding t_{emb} is taken as the standard sinosoidal positional embedding corresponding to timestep t. We then employ a series of transformer layers, incorporating multi-headed self-attention. We pass G along with the timestep embedding through these layers, enabling each point to capture dependencies with every other point as well as the timestep of the forward process. After the transformer layers, we use an MLP to map the D-dimensional embedding to the 2-dimensional observed noise ϵ_t:

$$H = \text{Transformer}(G, t_{\text{emb}}) \in \mathbb{R}^{(N+1)\times D}$$

The predicted noise from the denoising model is given by:

$$\hat{\epsilon_t} = \{\text{MLP}(H_1), \ldots, \text{MLP}(H_N)\} \in \mathbb{R}^{N\times 2}$$

IV. Experiments and Results

Our experiments in this section are specifically designed to address the following questions:

1) *How does ContourDiff compare to prior works for segmentation via diffusion and other segmentation approaches?*
2) *What forms of conditioning or guidance can be added to the diffusion model to improve the quality of freespace segmentation?*
3) *How can we sample efficiently from the diffusion model to enhance multimodal outputs and have better control over the generated samples?*

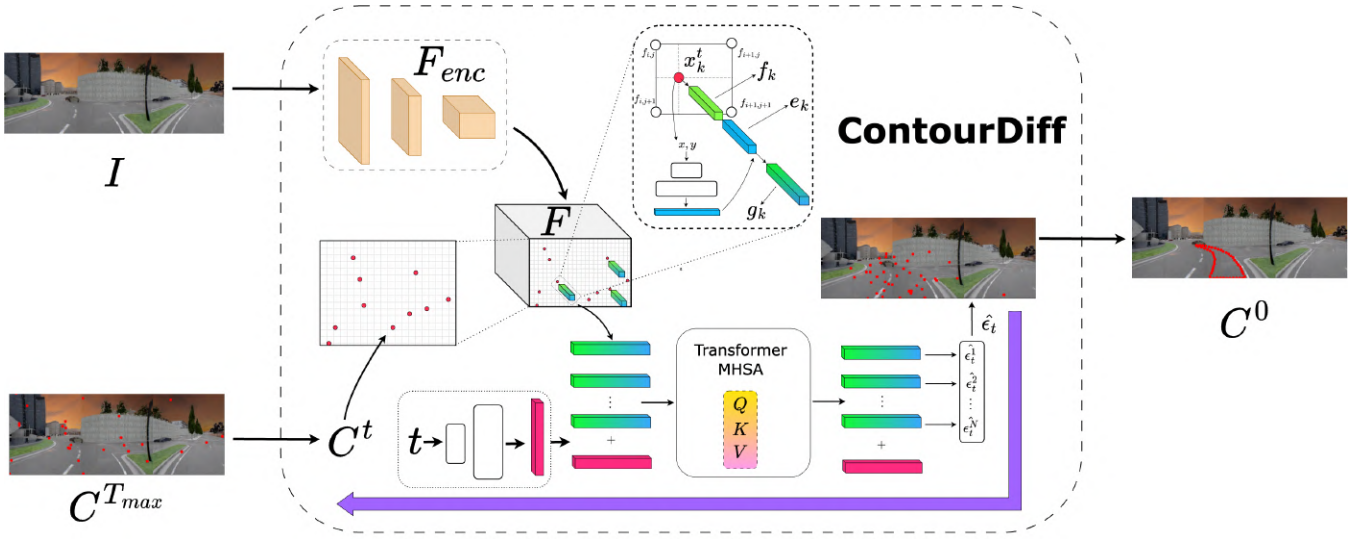

Fig. 5: **The architecture of the proposed ContourDiff.** The image I and the initial noisy contour $C^{T_{max}}$ are passed as input to the model. Note that C^t is visualized on top of image, and is not part of the image. The output of the model is the denoised contour C^0 which is obtained through running the reverse diffusion process.

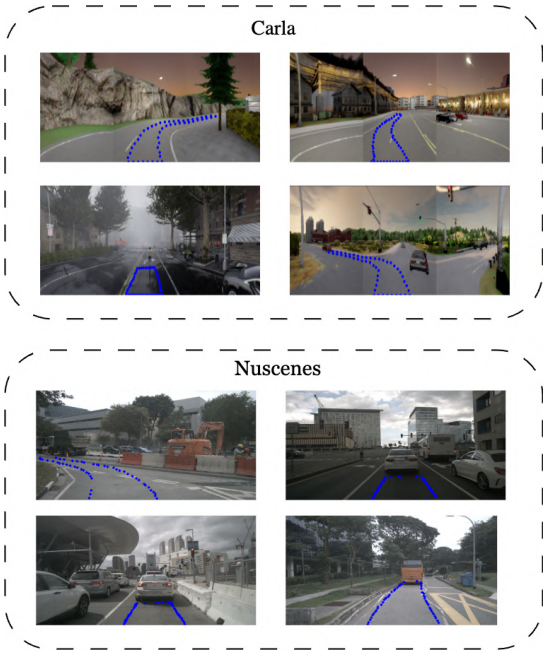

Fig. 6: **Training Samples :** We show different samples generated on CARLA and nuScenes, on applying the methodology described in III

A. Datasets

We evaluate and benchmark our model on two datasets: CARLA [23] and nuScenes [24]. Both datasets provide ego vehicle trajectory data, which we use to derive navigable free-space masks.

1) **CARLA:** We use the CARLA simulator to collect diverse training data using the LAV [25] data collection script. The dataset includes various driving scenarios such as straight roads, intersection turns, lane changes, and lane following. Our collected dataset comprises 82K frames from Towns 1–7, using three front-facing cameras with yaw angles of $-60°$, $0°$, and $60°$. We split the dataset into 75K frames for training and 7K frames for evaluation.
2) **nuScenes:** The nuScenes [24] dataset provides real-world urban driving scenarios with ground-truth ego trajectories. We use its official split, consisting of 700 sequences for training and 150 for validation.

B. Implementation Details

For CARLA, we stitch images from the three front-facing cameras, resulting in a final input size of 288×768. For nuScenes, the original image of size 900×1600 is resized to an input resolution of 256×512.

The model is trained with a learning rate of 10^{-4} and a batch size of 64 across four NVIDIA RTX 3080 Ti GPUs (effective batch size: 256). The forward diffusion process follows a cosine beta schedule with $T_{max} = 50$ timesteps. The model predicts $N = 50$ contour points, with features processed through six transformer blocks. Our empirical observations indicate that all validation metrics converge up until 50 training epochs.

TABLE I: **Quantitative Results for Freespace Generation.**

Method	CARLA			nuScenes		
	IoU (↑)	Obstacle Overlap (↓)	Off-Road Overlap (↓)	IoU (↑)	Obstacle Overlap (↓)	Off-Road Overlap (↓)
YOLOv11[26]	0.472	0.026	0.073	0.581	**0.006**	0.205
SegDiff[18]	0.676	0.052	**0.0032**	0.628	0.061	**0.022**
ContourDiff	**0.7707**	**0.0228**	0.0470	**0.687**	0.017	0.21

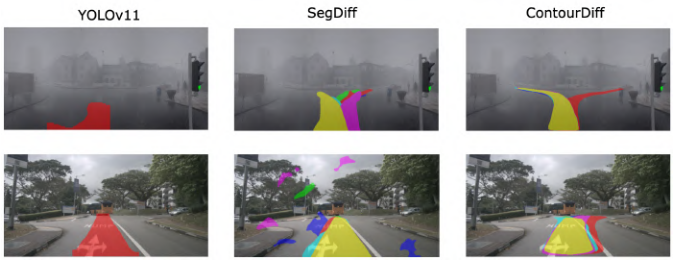

Fig. 7: **Above: CARLA** – Comparison of YOLOv11, SegDiff, and our proposed ContourDiff at an intersection. The non-generative baseline YOLOv11 struggles to predict the freespace segment. We present six samples from both SegDiff and ContourDiff, demonstrating that ContourDiff, with its prior of points and more efficient parameterization, produces more refined and reasonable segments. **Below: nuScenes** – YOLOv11 outputs only a single freespace segment. SegDiff fails to generate connected samples as it has no prior for doing so unlike the contour representation. Hence it often predicts disconnected masks which undermines the task of freespace prediction. In contrast, ContourDiff significantly improves freespace segmentation, producing more accurate and diverse results.

C. Evaluation Metrics

For freespace segmentation, we compute the mean **Intersection over Union (IoU)** between the predicted and ground truth freespace masks. To assess obstacle avoidance and safe navigation, we measure **Off-Road Overlap** which is the percentage of predicted freespace extending beyond the valid driving area and **Obstacle Overlap** which is the percentage of predicted freespace intersecting with detected obstacles.

For CARLA, to analyze multimodality, we introduce an additional metric called *Directional Deviation (DD)* to quantify variations in predicted samples. Specifically, we extract the centerline from the freespace mask by sampling the contour generated by the diffusion model. We then compute the angle of the line segment connecting the first and last points of the centerline. This process is repeated for six samples per image, and we calculate the mean and variance of these angles. Finally, we compute the average variance and average mean across the entire dataset to measure the diversity of the generated outputs. Additionally, we evaluate the Mean *Extent* of the angles, defined as the difference between the maximum and minimum angles among the six samples, providing further insight into the spread of predictions.

D. Comparing ContourDiff with other Segmentation Approaches

As mentioned before, since our goal is to predict front-view safe navigable contours using diffusion, we found no prior works that define freespace as corridors in the image frame. To the best of our knowledge, this problem remains unexplored in existing research. Therefore, we establish two baselines:

1) **Non Generative - YOLOv11:** We train a YOLOv11[26] segmentation model to demonstrate the limitations of a non-generative approach for this task.
2) **Generative - SegDiff:** To compare against a generative segmentation approach, we adopt SegDiff [18], a well-known diffusion-based image segmentation model which denoises over a standard mask based representation, as our baseline.

Table I presents a comparison of segmentation evaluation metrics between the baselines and our proposed ContourDiff. Since a non-generative supervised model can only predict a fixed set of masks deterministically given an input image, YOLOv11 struggles to capture the inherent variability in navigable contour prediction. The poor performance on CARLA and nuScenes further highlights the necessity of a generative model for this task. The generative baseline shows our model outperforming SegDiff in IoU and obstacle overlap, highlighting the benefits of contour-based predictions over segmentation masks. Notably, ContourDiff maintains low off-road overlap. In contrast, SegDiff often predicts no masks, lowering both validation IoU and obstacle overlap, resulting in a lower obstacle overlap than ContourDiff. Qualitative results are shown in Figure IV-B.

TABLE II: **Conditioning Results for Freespace Generation (CARLA).**

Method	IoU (↑)	Obstacle Overlap (↓)	Off-Road Overlap (↓)
ContourDiff	0.7707	0.0228	0.0470
Obstacle Guidance + ContourDiff	0.7653	0.0191	0.0436
Class Conditioned ContourDiff	0.6866	0.02513	0.055
Noise Template + ContourDiff	0.7613	0.0257	0.0519
Obstacle Guidance + Class Conditioned	0.6765	0.0239	0.0542

E. Enhancing Freespace Segmentation with Conditioning and Guidance

1) Class Conditioning-High Level Command Conditioning: We examine the effect of conditioning on a class token representing broader driving behaviors, such as lane changes or turns, on the generation of multimodal predictions. For every frame, we have a label corresponding to one of the 6 high level commands. Possible highlevel commands include turn-left, turn-right, go-straight, follow-lane, change-lane-to-left, change-lane-to-right. We one-hot encode the high-level command and project it to match the feature dimension of the transformer tokens. During training, we add this projected encoding to the set of input tokens. During inference, we sample a freespace segment for each of the six high-level commands, enabling diverse multimodal freespace predictions.

As shown in the Table II, the class-conditioned model generates contours that align with the expected driving behavior. However, since the ground truth represents only a single specific behavior, the model's diverse predictions—capturing multiple plausible behaviors—are evaluated against a single reference, leading to a lower validation IoU compared to the base model. Table III presents a comparison of Val IoU and DD for different ablations of the base model across various road scenarios in CARLA. Importantly, we see the effect of

TABLE III: **Multimodality evaluation of Freespace Generation Across Road Scenarios (CARLA).**

Method	NoLane				SingleLane				MultiLane				Intersection			
	IoU	DD			IoU	DD			IoU	DD			IoU	DD		
		Mean	Stddev	Extent		Mean	Stddev	Extent		Mean	Stddev	Extent		Mean	Stddev	Extent
Base Model	0.7919	98.35	3.12	7.5	0.7608	94.79	3.49	8.52	0.7739	96.06	2.95	7.26	0.7461	86.19	4.78	11.15
Noise Template	0.785	99.86	5.18	13.72	0.7522	94.23	5.18	13.53	0.7733	95.85	3.59	9.51	0.7017	88.43	11.24	27.48
Class Conditioning	0.7353	97.29	7.54	19.54	0.6863	94.39	7.06	18.55	0.7022	95.16	5.63	14.93	0.645	87.16	12.78	32.42
Obstacle Guidance	0.785	98.82	3.42	8.89	0.7573	95.5	3.51	9.36	0.7662	96.77	3.05	8.23	0.7206	87.65	5.48	14.04
Obstacle Guidance Class Conditioned	0.7342	98.83	7.73	19.92	0.6851	95.63	7.23	18.8	0.700	96.39	5.79	15.29	0.625	89.97	13.54	33.91

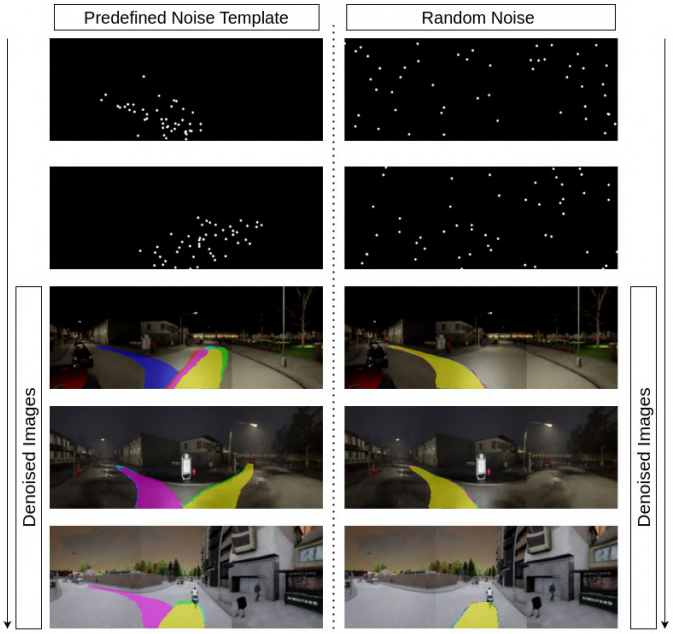

Fig. 8: **Effect of denoising from a set of predefined noise templates vs random noise template in the base model. Left:** Multimodality tends to increase as different modes are explored with different noise template initializations. **Right:** With random noise as initialization, the outputs tend to converge to a fixed sample.

class conditioning on the diversity of the freespace samples predicted in Table III, where we see a much higher extent in all cases than other methods.

2) Obstacle Guidance: We evaluate the role of obstacle masks in guiding freespace prediction. The diffusion model is encouraged to avoid predicting contours inside obstacle regions by applying a correction gradient to points that fall within obstacles. The correction gradient points outside the obstacle and forces the contour points to move outside the mask.

As demonstrated in Table II, obstacle guidance slightly reduces overlap with obstacles, ensuring that the predicted freespace aligns more closely with drivable regions.

F. Efficient Sampling for Enhanced Multimodal Generation

Motivated by the image editing technique in image diffusion models [27], where noise is added to an input image and then denoised through the diffusion model to produce an edited version, we investigate the impact of spatially varying noise patterns on generating well-defined multimodal results.

Figure 8 right presents samples generated starting from random noise alongside their corresponding starting noise. We observe that denoising from random noise often leads to convergence to fixed samples instead of exhibiting true multimodal behavior. This phenomenon is evident in Table III, where for the base model, the average extent at intersections, for example, is around 11 degrees.

To address this, we introduce structured initializations, allowing the model to reach local optima more effectively. As shown in Figure 8, we generate predefined noise templates by averaging K ground truth contours following a specific high-level command, K being a hyperparameter, and then applying the forward diffusion process for t timesteps. During denoising, we initialize from these templates and start the reverse denoising process from the same t timestep. The hyperparameter t is set to 10 in our experiments.

We find that initializing from six distinct noise templates, corresponding to six different high-level commands, significantly improves the multimodal behavior of the diffusion model while maintaining the other val metrics, as demonstrated in Table II and III, where both the extent and the variance are higher than that of the base model.

V. Conclusion

In this paper, we present a self supervised method for predicting visual corridors using diffusion models. Unlike previous approaches that rely on known obstacle locations, we treat the task as an image perception challenge, aiming to predict safe navigable contours directly from visual data. We introduce a self-supervised strategy for generating freespace samples by utilizing future ego trajectories and images. Additionally, we create a contour-based diffusion architecture that focuses on denoising contour points rather than employing a binary mask, resulting in outputs that are more structured and interpretable. We also perform comprehensive experiments on various conditioning strategies, guidance methods, and sampling techniques to improve multimodality and control over the generated samples. Our findings highlight the effectiveness of ContourDiff in generating diverse and precise freespace predictions, laying the groundwork for future research in generative methods for autonomous navigation.

References

[1] Dhruv Shah et al. "ViNT: A Foundation Model for Visual Navigation". In: *7th Annual Conference on Robot Learning*. 2023. URL: https://arxiv.org/abs/2306.14846 (cit. on p. 1).

[2] Penghao Wu et al. “Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 6119–6132 (cit. on p. 1).

[3] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. “Neat: Neural attention fields for end-to-end autonomous driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15793–15803 (cit. on p. 1).

[4] Jonah Philion and Sanja Fidler. “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, pp. 194–210 (cit. on p. 1).

[5] Wenyuan Zeng et al. “End-to-end interpretable neural motion planner”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 8660–8669 (cit. on p. 1).

[6] Yihan Hu et al. “Planning-oriented autonomous driving”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 17853–17862 (cit. on p. 1).

[7] Quang-Huy Che et al. “Twinlitenetplus: a stronger model for real-time drivable area and lane segmentation”. In: *arXiv preprint arXiv:2403.16958* (2024) (cit. on pp. 1, 2).

[8] Donghao Qiao and Farhana Zulkernine. “Drivable area detection using deep learning models for autonomous driving”. In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 5233–5238 (cit. on pp. 1, 2).

[9] Ciarán Hughes et al. “Drivespace: towards context-aware drivable area detection”. In: *Electronic Imaging* 31 (2019), pp. 1–9 (cit. on pp. 1, 2).

[10] Hengshuang Zhao et al. “Pyramid scene parsing network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890 (cit. on pp. 1, 2).

[11] Niklas Kochdumper and Stanley Bak. “Real-time capable decision making for autonomous driving using reachable sets”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 14169–14176 (cit. on pp. 1, 2).

[12] Gerald Würsching and Matthias Althoff. “Sampling-based optimal trajectory generation for autonomous vehicles using reachable sets”. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2021, pp. 828–835 (cit. on pp. 1, 2).

[13] Stefanie Manzinger, Christian Pek, and Matthias Althoff. “Using reachable sets for trajectory planning of automated vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 6.2 (2020), pp. 232–248 (cit. on pp. 1, 2).

[14] Piotr F Orzechowski, Kun Li, and Martin Lauer. “Towards responsibility-sensitive safety of automated vehicles with reachable set analysis”. In: *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE. 2019, pp. 1–6 (cit. on pp. 1, 2).

[15] Nivedita Rufus et al. “Grounding Linguistic Commands to Navigable Regions”. In: *CoRR* abs/2112.13031 (2021). arXiv: 2112.13031. URL: https://arxiv.org/abs/2112.13031 (cit. on p. 2).

[16] Naoki Hosomi et al. “Trimodal Navigable Region Segmentation Model: Grounding Navigation Instructions in Urban Areas”. In: *IEEE Robotics and Automation Letters* 9.5 (2024), pp. 4162–4169. DOI: 10.1109/LRA.2024.3376957 (cit. on p. 2).

[17] Dmitry Baranchuk et al. “Label-Efficient Semantic Segmentation with Diffusion Models”. In: *CoRR* abs/2112.03126 (2021). arXiv: 2112.03126. URL: https://arxiv.org/abs/2112.03126 (cit. on p. 2).

[18] Tomer Amit et al. “Segdiff: Image segmentation with diffusion probabilistic models”. In: *arXiv preprint arXiv:2112.00390* (2021) (cit. on pp. 2, 5, 6).

[19] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. “Scaling open-vocabulary object detection”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 72983–73007 (cit. on p. 3).

[20] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000) (cit. on p. 3).

[21] Satoshi Suzuki et al. “Topological structural analysis of digitized binary images by border following”. In: *Computer vision, graphics, and image processing* 30.1 (1985), pp. 32–46 (cit. on p. 3).

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851 (cit. on p. 4).

[23] Alexey Dosovitskiy et al. “CARLA: An open urban driving simulator”. In: *Conference on robot learning*. PMLR. 2017, pp. 1–16 (cit. on p. 5).

[24] Holger Caesar et al. “nuscenes: A multimodal dataset for autonomous driving”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631 (cit. on p. 5).

[25] Dian Chen and Philipp Krähenbühl. *Learning from All Vehicles*. 2022. arXiv: 2203.11934 [cs.RO]. URL: https://arxiv.org/abs/2203.11934 (cit. on p. 5).

[26] Rahima Khanam and Muhammad Hussain. “Yolov11: An overview of the key architectural enhancements”. In: *arXiv preprint arXiv:2410.17725* (2024) (cit. on pp. 5, 6).

[27] Chenlin Meng et al. “Sdedit: Guided image synthesis and editing with stochastic differential equations”. In: *arXiv preprint arXiv:2108.01073* (2021) (cit. on p. 7).