MapDiffusion: Generative Diffusion for Vectorized Online HD Map Construction and Uncertainty Estimation in Autonomous Driving

Thomas Monninger[1,2], Zihan Zhang[3,*], Zhipeng Mo[1], Md Zafar Anwar[1], Steffen Staab[2,4], Sihao Ding[1]

***Abstract*— Autonomous driving requires an understanding of the static environment from sensor data. Learned Bird's-Eye View (BEV) encoders are commonly used to fuse multiple inputs, and a vector decoder predicts a vectorized map representation from the latent BEV grid. However, traditional map construction models provide deterministic point estimates, failing to capture uncertainty and the inherent ambiguities of real-world environments, such as occlusions and missing lane markings. We propose MapDiffusion, a novel generative approach that leverages the diffusion paradigm to learn the full distribution of possible vectorized maps. Instead of predicting a single deterministic output from learned queries, MapDiffusion iteratively refines randomly initialized queries, conditioned on a BEV latent grid, to generate multiple plausible map samples. This allows aggregating samples to improve prediction accuracy and deriving uncertainty estimates that directly correlate with scene ambiguity. Extensive experiments on the nuScenes dataset demonstrate that MapDiffusion achieves state-of-the-art performance in online map construction, surpassing the baseline by 5% in single-sample performance. We further show that aggregating multiple samples consistently improves performance along the ROC curve, validating the benefit of distribution modeling. Additionally, our uncertainty estimates are significantly higher in occluded areas, reinforcing their value in identifying regions with ambiguous sensor input. By modeling the full map distribution, MapDiffusion enhances the robustness and reliability of online vectorized HD map construction, enabling uncertainty-aware decision-making for autonomous vehicles in complex environments.**

I. Introduction

The safe operation of an autonomous driving system requires an accurate and complete representation of the static infrastructure surrounding the vehicle. This map representation must be derived in real-time from sensor information (*i.e.*, online map construction) to react to the current real-world scenario. Also, most autonomous driving systems require the map in vectorized form for use in their planning system, since a vectorized representation provides instance-level information and spatial consistency [1-3] However, the task of vectorized online map construction is inherently challenging due to the ambiguity of the real world. A wide lane may be a single lane or two lanes with missing markings, intersections often lack explicit lane definitions, and construction zones introduce temporary changes that may contradict the original lane layout. Additionally, occlusions caused by other vehicles or roadside obstacles increase ambiguity. Committing to a single interpretation in such scenarios can be misleading and potentially unsafe.

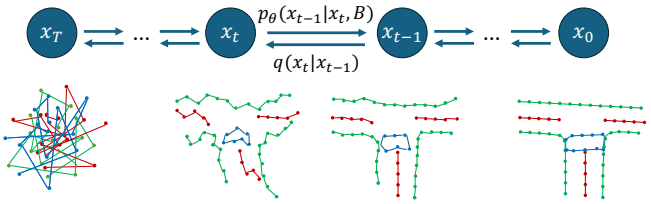


Fig. 1: Overview of the diffusion forward and backward processes on vectorized HD maps used in MapDiffusion.

Previous approaches to online mapping perform deterministic construction of a map from the provided sensor data. For this, they commonly use learned Bird's-Eye View (BEV) encoders to fuse information from multiple camera views into a joint latent space. For the decoder, initial approaches focused on predicting a raster map representation [4-7], while more recent approaches directly predict vectorized representations [1, 2, 8]. In both cases, previous deterministic models commit to a single interpretation for a given input. In an ambiguous traffic scene, this map prediction may be incorrect, leading to unsafe decisions. We argue that capturing the full distribution is required to consider all plausible map configurations in downstream decision-making. While a few works investigate diffusion for map construction, unlike our work, they either operate on raster representations [9, 10] or just do refinement of initial proposals [11]. Instead, we propose the use of full generative diffusion for vectorized High-Definition (HD) map construction. We introduce a novel graph diffusion decoder that denoises randomly initialized queries conditioned on camera features from the latent BEV grid. MapDiffusion learns the full vector map distribution and therefore can generate plausible samples to capture the ambiguity of the real world. Furthermore, we show that the variance between the sampling results can serve as a measure of uncertainty in the constructed online map, providing an effective way to capture ambiguity in the real world.

The primary contributions of this paper are as follows:

- We propose the use of generative diffusion for the task of online vectorized HD map construction and implement a new model, MapDiffusion, that performs denoising conditioned on a latent BEV grid.
- We investigate the variance of the sampling output by interpreting it as a measure of uncertainty and find a significant increase of 31 % in occluded areas.

[1]Mercedes-Benz Research & Development North America, Sunnyvale, CA, USA (email: thomas.monninger@mercedes-benz.com)
[2]University of Stuttgart, Institute for Artificial Intelligence, Stuttgart, Germany (email: steffen.staab@ki.uni-stuttgart.de)
[3]University of California, San Diego, La Jolla, CA, USA
[4]University of Southampton, Southampton, United Kingdom
*Work was done during an internship at Mercedes-Benz Research & Development North America.

- We conduct extensive experiments on the nuScenes dataset, demonstrating a 5 % relative improvement in single-sample performance and further improvements by aggregating multiple samples.

II. Related Work

A. Online HD Map Construction

In autonomous driving, static elements such as roads, lane dividers, and pedestrian crossings are typically represented in a map. Recent work has made notable progress in constructing a map representation online directly from sensor data. Typically, a learned BEV encoder is used to fuse information from multiple camera views into a joint representation. The first works on learned BEV encoders predict a raster map by treating it as a segmentation task, *i.e.*, a pixel-wise classification of map elements [4-7].

Starting with VectorMapNet [1], more recent approaches construct end-to-end HD maps by directly predicting the vectorized map elements. MapTR [12] addresses the ambiguity of selecting a discrete set of points to model geometries. It employs permutation-equivalent modeling to stabilize the learning process. StreamMapNet [8] uses a powerful 6-layer transformer decoder that performs temporal aggregation by streaming queries from the previous frame. AugMapNet [13] adds dense spatial supervision for improved structure of the latent space. SQD-MapNet [14] injects a few noise-perturbed GT queries from the previous frame during training. A query denoising module is added to improve temporal consistency. We use the general idea of query denoising in the context of a diffusion framework. In contrast, during our training, we only use noise-perturbed GT queries from the current frame and execute the full decoder iteratively to learn sampling from random Gaussian noise.

B. Diffusion Models

Generative modeling can generate complex objects in a domain, *e.g.*, high-fidelity image synthesis, video generation, and natural language processing [15-18]. The diffusion process can be controlled by conditioning on additional information [19, 20]. We leverage generative modeling to sample from a probability distribution over vectorized maps conditioned on sensor data encoded in a BEV grid B.

Denoising Diffusion Probabilistic Models (DDPMs) [15] model a Markovian forward process, q, that transforms x into Gaussian noise over multiple diffusion time steps t. Figure 1 visualizes this for a vectorized HD map with x_0 being the vectorized map and x_T being polylines with random coordinates. A denoising prediction network with weights θ learns the reverse process $p_\theta(x_t, t, B)$, where the latent BEV grid B serves as condition to guide the prediction. Once trained on a distribution, the diffusion model can generate samples from that distribution.

C. Diffusion Models for Mapping

As a first application, the diffusion paradigm has been used to generate 3D occupancy maps [21-24], capturing the three-dimensional geometric structure of the surroundings. Other works [25-27] use diffusion models conditioned on aerial images or other geospatial context to generate semantic map layers for various use cases. More recently, works have begun to explore diffusion models to construct online raster maps from on-road camera views. DiffMap [9] leverages a latent diffusion model and enhances the generated raster map by integrating structured priors inherent in map segmentation masks. DifFUSER [10] extends the diffusion paradigm to both 3D object detection and raster map prediction. In contrast to the above works, our approach applies the diffusion paradigm to directly predict vectorized map elements.

A new research topic is diffusion-based generation of vector representations. DiffusionDet [18] and DiffBEV [28] apply the diffusion paradigm to object detection and generate vectorized bounding boxes. They condition the diffusion process on the image and BEV space, respectively. House-Diffusion [29] performs diffusion to generate vectorized floor plans, using similar techniques to our work, but for a different learning task. To our knowledge, the only work that uses diffusion for online vectorized map construction is PolyDiffuse [11]. Its Guided Set Diffusion Model uses a guidance network to manage noise injection and maintain unique representations for the diffusion model, enabling accurate polygonal shape reconstruction of floor plans and HD maps. However, they use diffusion as a refinement step on top of coarse predictions from existing map construction models, correcting structural errors and enhancing the accuracy of the predicted polylines. Unlike MapDiffusion, their overall accuracy remains heavily dependent on the performance of the baseline model, and their method cannot be used to generate diverse samples from the learned distribution.

D. Uncertainty Estimation

Gu *et al*. [30] extend methods for online map construction with uncertainty estimation. Instead of predicting the vectorized coordinates, they predict the parameters of a Laplace distribution for each polyline point. To show the benefit of predicting a map distribution, they evaluate its use in a trajectory prediction model and find up to $15\,\%$ improved prediction performance. We follow this argument and produce uncertainty estimates from the sampling variance of our diffusion process. By considering multiple samples, we not only rely on one set of predicted polylines, yielding denser spatial uncertainty estimates.

Diffusion models have successfully been used for uncertainty estimation in other domains. CARD [31] is a diffusion-based approach that uses conditional generative models to uncover predictive distributions, hence capturing the uncertainty. Du and Li [32] also use diffusion for uncertainty estimation and apply this to active domain adaptation.

In the context of trajectory prediction, uncertainty is inherently present in the task due to its multi-modal distribution. MotionDiffuser [33] uses controllable diffusion to sample plausible trajectories. To the best of our knowledge, we are the first to leverage diffusion-based sampling to estimate the uncertainty of online map construction.

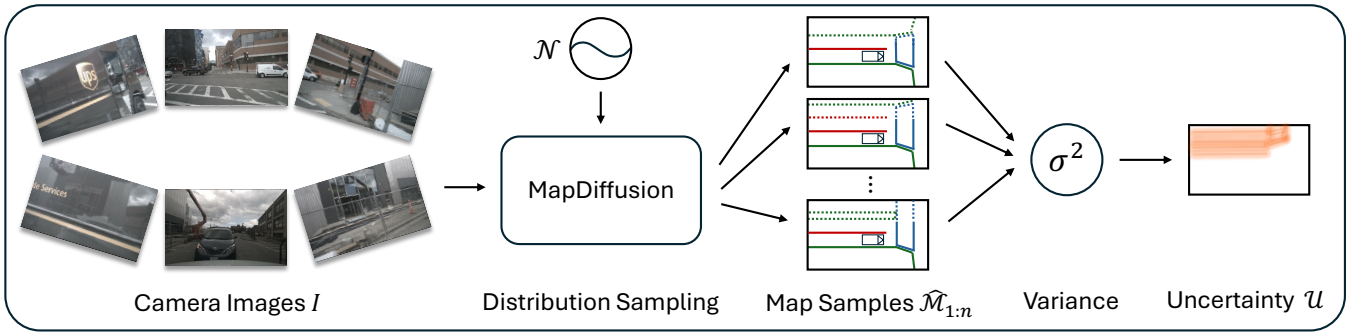


Fig. 2: Illustration of MapDiffusion approach based on schematic traffic scene with an occluded camera view on the ego's left. MapDiffusion uses generative diffusion to predict samples of map distribution from camera images. The samples show plausible predictions for the occluded area on the ego's left (differences are indicated with dashed lines). The variance across samples is used as spatial uncertainty estimation, yielding high uncertainty in the occluded area (in orange on the right).

III. Approach

A. Problem Statement

Let $I = \{i_1, \ldots, i_m\}$ be the set of image frames from the m monocular cameras mounted on the ego vehicle. Moreover, for a given scene, let $\mathcal{P}_{\text{div}}$, $\mathcal{P}_{\text{bound}}$, and $\mathcal{P}_{\text{ped}}$ be the sets of polylines representing lane dividers, lane boundaries, and pedestrian crossings, respectively, with a polyline, $P = [(x_j, y_j)]_{j=1}^{N_P}$, being a sequence of N_P points. Let $\mathcal{M} = (\mathcal{P}_{\text{div}}, \mathcal{P}_{\text{bound}}, \mathcal{P}_{\text{ped}})$ be the local HD map with the ego vehicle at the origin. The goal is to find a function f that returns an estimate of the local HD map, $\hat{\mathcal{M}}$, for a given set of image frames I, *i.e.* $\hat{\mathcal{M}} = f(I)$. Additionally, the goal is to provide a function $\mathcal{U}$ that provides an uncertainty estimate for $\hat{\mathcal{M}}$ at a Cartesian location $(x, y) \in \mathbb{R}^2$. This uncertainty estimate provides a fuzzy, qualitative indicator of confidence for the predicted map at each location in the scene based on the perceived ambiguity at that location.

B. MapDiffusion

We propose MapDiffusion, a novel model that leverages generative diffusion to sample map predictions. An overview of our approach is shown in Figure 2. MapDiffusion can generate samples from the learned map distribution $\mathcal{M}$. The variance across generated map samples $\hat{\mathcal{M}}$ serves as spatial uncertainty estimate $\mathcal{U}$. In the figure, the camera view to the left is blocked by a delivery truck. MapDiffusion can sample plausible map configurations with high variance in the occluded area. Consequently, $\mathcal{U}$ suggests a high uncertainty in that area. The following sections cover various aspects of our MapDiffusion approach.

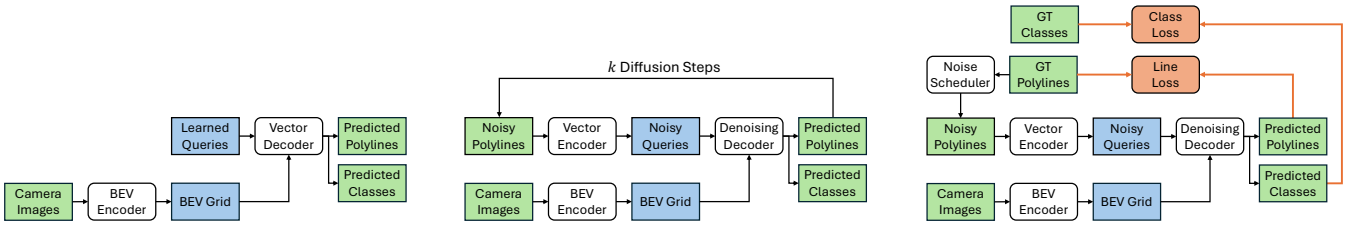
1) Model Architecture: MapDiffusion uses StreamMapNet [8] as reference architecture; the high-level architecture of both models is shown in Figure 3. Both use a learned BEV encoder to generate a latent representation of the camera features. We leverage a DETR-style transformer decoder [34] that performs query refinement conditioned on the latent representation of the BEV grid. We adapt the decoder to a diffusion framework such that the denoising decoder starts with random polylines as input and progressively performs denoising through an iterative refinement process.

2) Training Process and Noise Scheduler: Figure 3c shows an overview of the training process. During training, a noise scheduler performs the forward process q. The Noise Scheduler determines the noise added to the GT at each diffusion time step t. As visualized in Figure 1, we perform q and p_θ in vector space. The denoising decoder uses an embedding of the time step t as an additional input to condition its prediction on the noise step. It is trained to minimize the error between $p_\theta(q(x_0, t), t, B)$ and x_0 for all $t \in [0, T]$, which is visualized "Line Loss" in Figure 3c. Prediction of the class score is excluded from the diffusion process and instead performed in the final step, which enables diffusion to happen purely in the vector space. The overall training is a joint optimization of the diffusion process and the classification task.

3) Diffusion Conditioning: MapDiffusion conditions the diffusion process on a latent BEV grid B to guide the denoising process by camera features. This ensures that the generated map prediction is not only a plausible map, but also consistent with what is visible in the camera images I. To integrate this additional context, we employ deformable cross-attention [35] to have the denoising decoder query the latent BEV grid.

4) Query Padding: The DETR-style decoder [34] operates with a fixed number of l queries, each corresponding to either a map element in the output, or the "no object" class. During the training of MapDiffusion, the GT map elements serve as a starting point, and the forward process q is applied to initialize the queries. However, the GT contains a variable number of map elements, typically fewer than l, so padding is required for the queries. Padding with Gaussian noise performs best based on our ablation study (see Section IV-G.3).

5) Temporal Aggregation in Decoder: MapDiffusion uses the BEV-space temporal aggregation on the encoder side from StreamMapNet [8]. StreamMapNet additionally uses temporal aggregation in the vector decoder by incorporating refined queries from the previous temporal step into the second layer of the decoder. In our diffusion decoder, we opt not to use the refined queries from the previous temporal step



(a) StreamMapNet (b) MapDiffusion (Inference) (c) MapDiffusion (Training)

Fig. 3: Overview of StreamMapNet, MapDiffusion during inference, and MapDiffusion during training, with box colors indicating explicit representations (green), latent representations (blue), computation modules (white), and losses (orange).

so that map predictions are generated purely from random noise. While this misses out on past object-space information to improve the consistency of the predicted map elements over time, it simplifies the architecture and preserves the original concept of diffusion.

6) Sampling: The sampling process follows Denoising Diffusion Implicit Model (DDIM) [36], which allows generating high-quality map predictions with only a few diffusion steps. The sampling process is shown in Figure 3b. The BEV encoder at the bottom is only calculated once for efficiency. The resultant latent BEV grid is used to condition the denoising decoder. The denoising decoder refines the vectorized polyline prediction through k diffusion steps. Additionally, filtering of the output is performed during inference. Specifically, outputs with a score below τ are dropped and randomly initialized for the next inference step. This prevents the model from having to deal with suboptimal initializations.

C. Aggregating Samples for Refined Prediction and Uncertainty Estimation

During inference, for a given set of camera images I, we sample n map predictions from the map distribution. We use different aggregation strategies to get to a refined prediction and to generate an uncertainty estimation. To get the distribution of the predicted local HD map, we generate n samples of map predictions, denoted as $\hat{\mathcal{M}}_i$, for $i \in [1, 2, \ldots, n]$. It is important to note that the number of samples n is different from the number of diffusion steps k performed to generate one sample. We present an aggregation strategy to get a refined prediction as well as an uncertainty estimate.

1) Sample Aggregation as Refined Prediction: We aggregate n samples to obtain a refined prediction from a map distribution. Since the aggregation of sets of polylines is non-trivial, we perform aggregation in raster space, which is sufficient to demonstrate our point. For each predicted sample i from n samples and class $c \in C$, let $\hat{\mathcal{P}}_c^i$ denote the set of predicted polylines corresponding to class c and let $\mathcal{S}_c^i$ represent the associated confidence scores. A polyline indexed by j, $\hat{\mathcal{P}}_c^i[j]$, is converted into a rasterized map, and each location is weighted by its corresponding score $\mathcal{S}_c^i[j]$. All polylines for class c are summed to produce the weighted raster $\mathcal{R}_c^i \in \mathbb{R}^{H \times W}$:

$$\mathcal{R}_c^i(x, y) = \sum_{j=1}^{|\hat{\mathcal{P}}_c^i|} \mathcal{S}_c^i[j] \, \text{Rasterize}(\hat{\mathcal{P}}_c^i[j]). \tag{1}$$

Aggregation of the class probability distributions requires spatial smoothness, so a Gaussian kernel, $G \in \mathbb{R}^{g \times g}$, is applied to the weighted raster $\mathcal{R}_c^i$. Due to potential overlap of different polylines, the resulting values are clipped to the range $[0, 1]$ to obtain a class probability map $\mathcal{D}_c^i$:

$$\mathcal{D}_c^i(x, y) = \min(1, G * \mathcal{R}_c^i(x, y)), \tag{2}$$

where $*$ denotes the convolution operation.

The aggregated class probability distribution is given by the mean probability per class at location (x, y) as:

$$\mathcal{D}_c(x, y) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{D}_c^i(x, y). \tag{3}$$

To generate a refined prediction from the distribution, the scores are thresholded with a binarization threshold b.

2) Sample Variance as Uncertainty Estimation: According to the problem statement, we need an uncertainty estimation that captures the ambiguity of the real world. Our diffusion model can generate diverse samples that capture the multi-modality of the map distribution. Given n samples from our model for a given set of images I, we can leverage the sample variance to compute a spatial uncertainty. For simplicity, we construct an uncertainty estimate at each spatial location (x, y) of a predefined grid, $\mathcal{U} \in \mathbb{R}^{H \times W}$, based on the total variance across class scores at that location. Specifically, we first compute the per-location variance, σ_c^2, by calculating the variance across the n samples of $\mathcal{D}_c^i$. Then we compute the per-location uncertainty map by summing the variances across all classes:

$$\mathcal{U}(x, y) = \sum_{c \in C} \sigma_c^2(x, y). \tag{4}$$

This formulation provides a spatially-resolved uncertainty estimate, where higher values of $\mathcal{U}$ indicate greater variability in class predictions across samples, highlighting regions of increased uncertainty in the local HD map. The values of $\mathcal{U}$ are derived from variance measures and are not normalized; consequently, they should be interpreted as qualitative indicators.

IV. Experiments

A. Dataset and Evaluation Metrics

We conduct our experiments on the nuScenes dataset [37], which provides data points at 2 Hz. Each data point includes images from six monocular cameras, I, and vectorized GT sets (*i.e.*, $\mathcal{P}_{\mathrm{div}}$, $\mathcal{P}_{\mathrm{bound}}$, and $\mathcal{P}_{\mathrm{ped}}$ for lane dividers, lane boundaries, and pedestrian crossings, respectively). We use the StreamMapNet training and validation split without geospatial overlaps [8]. The graph decoding task is evaluated using Average Precision (AP) and mean Average Precision (mAP). For single-sample and n-sample rasterized predictions, we measure the True Positive Rate (TPR) and False Positive Rate (FPR) for various operational points and compare them using the Receiver Operating Characteristic (ROC) curve, and Area Under the Curve (AUC). Inference time is measured in Frames Per Second (FPS).

B. Experimental Setup

We adopt the StreamMapNet training configuration with 24 epochs and a batch size of 1. The model is trained in parallel on 8 NVIDIA V100 GPUs. For optimization, we employ the AdamW optimizer with a cosine annealing schedule and a learning rate of 2×10^{-4}. The dimensions of the BEV grid are set to 100×50, covering a perception range of $60\,\mathrm{m} \times 30\,\mathrm{m}$. For diffusion, we choose a cosine noise scheduler and set $T = 1000$. Inference uses $\eta = 0.5$ and $k = 5$ diffusion steps. For experiments generating multiple samples, we set $n = 10$. The $\mathrm{Rasterize}$ operator uses pixel width 1 on polylines with prediction score > 0.4. The Gaussian filter uses $g = 5$ ($3\,\mathrm{m}$) and $\sigma = 1$ ($0.6\,\mathrm{m}$).

C. Baseline Models

We use StreamMapNet [8] as reference architecture and primary baseline. Other baselines include PolyDiffuse [11], the only diffusion-based method for online vectorized map construction, SQD-MapNet [14], which performs a similar strategy of denoising on queries, and common methods for vectorized map construction, including VectorMapNet [1], MapTR [12], MapTRv2 [2], MapVR [38], and MGMap [39].

D. Quantitative Results of Model

Table I shows the qualitative results. MapDiffusion reaches $35.6\,\%$ mAP, a $5.3\,\%$ relative improvement over the StreamMapNet baseline with $33.8\,\%$ mAP. Notably, this performance is achieved despite MapDiffusion operating without learned queries and lacking temporal aggregation in the decoder, underscoring its efficacy in generating high-quality map samples under these constraints. The model remains highly efficient, achieving real-time performance at 8.0 FPS with five diffusion steps. In a single-step configuration, it matches StreamMapNet with 12.8 FPS. MapDiffusion also outperforms common baselines including VectorMapNet [1], MapTR [12], MapTRv2 [2], MapVR [38], and MGMap [39]. We run the public implementation of SQD-MapNet [14] on the new nuScenes split with batch size 1 and get $33.1\,\%$ mAP, which ranks it below our MapDiffusion approach. PolyDiffuse [11] did not release code to reproduce the results on the new nuScenes split, but they report a result below StreamMapNet on the original split. Given our $5.3\,\%$ relative improvement over StreamMapNet, we assume from transitive reasoning that MapDiffusion outperforms PolyDiffuse. Very recent works reach beyond the performance of StreamMapNet (*e.g.*, MapQR [40], HIMap [41]), but do not report results on the new nuScenes split. This work intends to show the general possibility of using diffusion for generative map construction and the benefit of deriving an uncertainty estimate. Most importantly, our paradigm can be applied to these works as well.

Table I: Performance of MapDiffusion compared to various baselines at perception range $60\,\mathrm{m} \times 30\,\mathrm{m}$ on nuScenes split without geospatial overlap [8]. * results from [8], all other results are reproduced. AP thresholds are $\{0.5, 1.0, 1.5\}$.

Method	$\mathrm{AP}_{\mathrm{ped}}$	$\mathrm{AP}_{\mathrm{div}}$	$\mathrm{AP}_{\mathrm{bound}}$	mAP
VectorMapNet* [1]	15.8	17.0	21.2	18.0
MapTR [12]	7.5	23.0	35.8	22.1
MapVR [38]	10.1	22.6	35.7	22.8
MGMap [39]	7.9	25.6	37.4	23.7
MapTRv2 [2]	16.2	28.7	44.8	29.9
SQD-MapNet [14]	31.6	27.4	40.4	33.1
StreamMapNet [8]	31.2	27.3	**42.9**	33.8
MapDiffusion (ours)	**32.9**	**31.4**	42.4	**35.6**

E. Aggregating Samples

All previous results were calculated from one sample. Experiments below show the benefit of multiple samples.

1) Refined Prediction: Using the rasterized predicted class distribution $\mathcal{D}_c$ and the rasterized GT map $\mathcal{M}$, we compute the True Positive Rate and False Positive Rate for different binarization thresholds b. The resulting ROC curves for $n = 1$ and $n = 10$ are visualized in Figure 4. The aggregated prediction from 10 samples is strictly better along the curve, confirming the hypothesized benefit of sampling multiple predictions from the distribution. Accordingly, the AUC is 0.89 for $n = 1$ and 0.92 for $n = 10$, indicating a $3.4\,\%$ relative improvement from aggregating multiple samples. Please note that while this improvement is notable, we perform this evaluation primarily to show the information gain by generating multiple samples from the full distribution. The true value lies in considering all plausible map configurations in the downstream planning module for more robust and uncertainty-aware decision-making.

2) Uncertainty and Visibility: We aim to show that the uncertainty maps $\mathcal{U}$ from our approach correspond to ambiguity in the real world. In that case, $\mathcal{U}$ is expected to estimate high uncertainty for areas that are not visible, for example, due to a vehicle occluding the camera's field of view. We compute the spatial relation between our estimated uncertainty maps and GT visibility masks. Since nuScenes does not provide visibility masks directly, we generate them from the Occ3D dataset [42], which provides occupancy maps for the nuScenes dataset. We exclude the ground layer and occupancy of type "flat" and project the 3D voxels into 2D BEV based on whether any of the voxels in the

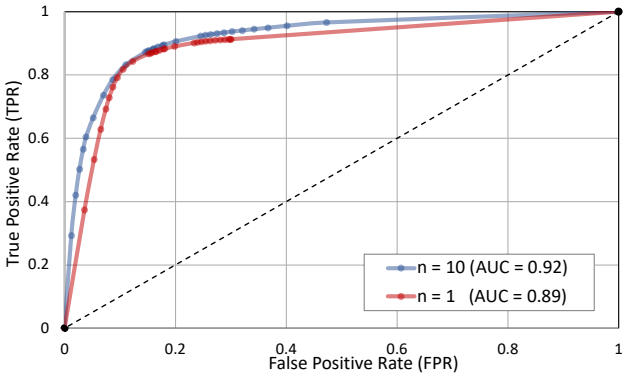


Fig. 4: ROC curves for one sample (red) and 10 aggregated samples (blue). Dashed line is random classifier. AUC indicates that aggregated map is better than single-sample map.

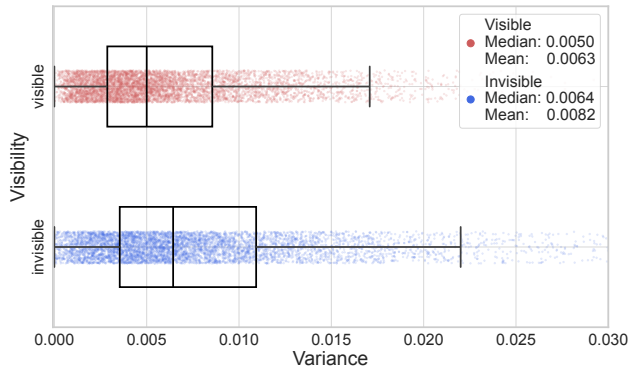


Fig. 5: Relation between visibility and variance across predicted map samples, which we use as an uncertainty estimate.

z direction are occluded. Finally, we perform ray-tracing on the 2D projection to calculate our visibility masks. For our evaluation, we calculate the mean uncertainty per traffic scene separately for visible areas and invisible areas and compare them. We consider pixels that are visible (based on the visibility map) or part of the drivable road surface (based on the dense raster GT). The distribution of the variances, which we use as uncertainty estimates, is shown in Figure 5. The mean uncertainty for the visible area is 0.0063. For the invisible area, it is 0.0082, which is 31 % higher. We conduct a one-sided t-test and find that uncertainty in invisible areas is significantly higher than in visible areas ($\alpha = 0.01$, $p < 0.001$). It is important to note that the MapDiffusion model performs temporal aggregation in BEV space and hence has access to more spatial features than currently visible, reducing the uncertainty in invisible areas that were visible in previous time steps. Therefore, the relation between uncertainty and visibility is assumed to be even higher for a model with no temporal aggregation.

F. Qualitative Results

Figure 6 shows qualitative results for two traffic scenes. Multiple map construction samples are visualized for each traffic scene to illustrate the sampling variance. The predicted map demonstrates high accuracy in visible areas, validating its state-of-the-art quantitative performance. In both scenes, the variance across sampled map predictions is high in occluded areas, showcasing the relation between uncertainty and perceptual ambiguity.

Table II: Ablation on Diffusion Parameters k, η, τ.

Steps k	η	τ	FPS	AP_{ped}	AP_{div}	AP_{bound}	mAP
1	0.5	0.5	12.8	32.3	30.8	42.5	35.2
2	0.5	0.5	11.1	32.3	31.4	42.5	35.4
3	0.5	0.5	9.9	32.9	31.2	42.5	35.5
4	0.5	0.5	8.8	32.9	31.5	42.5	**35.6**
5	0.5	0.5	8.0	32.9	31.4	42.4	**35.6**
5	0.1	0.5	8.0	32.7	31.3	42.3	35.4
5	0.3	0.5	8.0	32.6	31.2	42.3	35.4
5	0.5	0.5	8.0	32.9	31.4	42.4	**35.6**
5	0.7	0.5	8.0	32.6	31.4	42.4	35.5
5	0.9	0.5	8.0	32.5	31.3	42.4	35.4
5	0.5	0.1	8.0	30.9	27.8	39.3	32.7
5	0.5	0.3	8.0	32.5	31.2	42.3	35.3
5	0.5	0.5	8.0	32.9	31.4	42.4	**35.6**
5	0.5	0.7	8.0	33.0	31.1	42.3	35.5
5	0.5	0.9	8.0	32.7	30.6	42.1	35.1

Table III: Ablation on query padding strategies. Models were trained for 12 epochs with a pretrained, frozen BEV encoder.

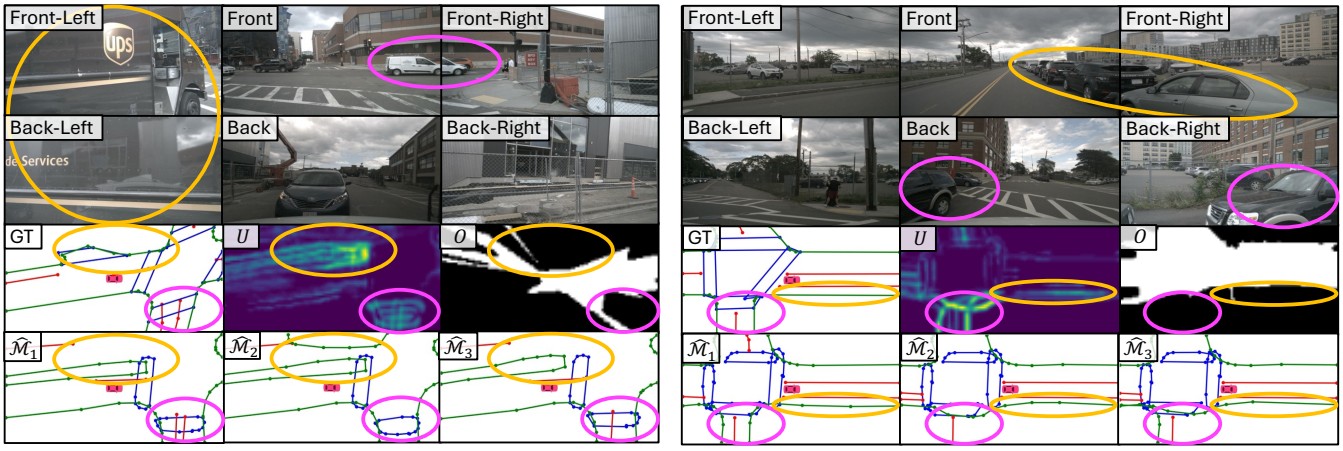
Padding	AP_{ped}	AP_{div}	AP_{bound}	mAP
Repeat	23.6	22.8	35.8	27.4
Zero	25.2	28.3	37.8	30.5
Smooth	24.8	27.8	38.1	30.2
Gaussian	26.3	27.8	38.6	**30.9**
Uniform	25.0	27.7	37.3	30.0

G. Ablation Studies

We perform ablation studies to examine the efficacy of design decisions, including choice of diffusion parameters, pretraining of the BEV encoder, and padding strategies.

1) Diffusion Parameters: We evaluate the number of diffusion steps, the η parameter in DDIM sampling [36], and the query threshold τ. The results are shown in Table II. For the number of steps, we see better performance on the map construction task with more diffusion steps, saturating at around 5. The number of diffusion steps has an effect on the runtime since it requires sequential execution of the denoising module, a typical downside of diffusion models. Our approach addresses this issue by excluding the learned BEV encoder from the diffusion process, so the latent BEV grid only has to be computed once. On an NVIDIA A10 GPU, performing an additional denoising step adds around 12 ms. This is a +15 % increase from the baseline inference time that has just one decoder pass. The five diffusion steps necessary to achieve saturated performance increase the total time by 60 %, achieving 8.0 FPS. It is important that only the number of diffusion steps k increases the latency. Sampling the distribution with n predictions can be done in parallel.

The η parameter has only a minor influence, and we get the best results for $\eta = 0.5$. The query threshold τ, which determines which queries are kept for the next diffusion



(a) Occlusion from the delivery truck on the ego's left is shown with orange circles and results in different bound predictions. Occlusion from the white van on the ego's front-right is shown with pink circles and results in varying div and ped predictions.

(b) Occlusion from parked cars on the ego's right is shown with orange circles and results in different bound predictions. The occluded intersection at the ego's back-right is shown with pink circles and results in varying bound and ped predictions.

Fig. 6: Two qualitative results of MapDiffusion. The top two rows show the 6 camera views. The third row shows the GT (left), uncertainty $\mathcal{U}$ (center), and occlusion (right) maps. The bottom rows show 3 predicted samples $\hat{\mathcal{M}}_{1:3}$, where green is bound, red is div, and blue is ped.

step, has a stronger impact. Keeping almost all queries ($\tau = 0.1$) has the worst performance with 32.7 % mAP. The best performance reaches 35.6 % mAP for $\tau = 0.5$. Dropping most queries ($\tau = 0.9$) degrades performance again to 35.1 % mAP. Overall, given an mAP of around 35 % for most settings, we find that MapDiffusion is robust to hyperparameter choices. While even one diffusion step already reaches a good result, more steps are expected to increase sample variance, which is beneficial for capturing the full distribution and also generating $\mathcal{U}$. Based on the ablation results, we choose the number of diffusion steps to be 5, $\eta = 0.5$, and $\tau = 0.5$.

2) Pretraining of BEV Encoder: We assess the benefit of pretraining the BEV encoder. First, we train the full MapDiffusion model with randomly initialized weights, reaching 35.6 % mAP. We then train a new MapDiffusion model with the frozen pre-trained BEV encoder. While the optimization is faster with a pre-trained BEV encoder (25.7 % mAP vs. 17.5 % mAP after 6 epochs, and convergence around 12 epochs), the resulting model reaches only 31.7 % mAP. Hence, we opt for training from scratch for all final experiments. We use the pretraining method exclusively for ablations, such as ablating the padding strategy, and train for 12 epochs there for efficiency reasons.

3) Padding Strategy: We compare the following strategies for query padding. "Repeat": repeats existing polylines, "Zero": zero values for all additional polylines, "Smooth": smooth random polylines (both straight and curved) or polygons (*e.g.*, elliptical shapes), "Gaussian": Gaussian noise with $\mu = 0.5$ and $\sigma = 0.25$ clipped to [0,1] (range of normalized GT), and "Uniform": uniform noise with the boundaries $[0, 1]$. For efficiency, we train them with a pre-trained BEV encoder for 12 epochs (see Section IV-G.2). The results are shown in Table III. "Gaussian" performs best. "Zero", "Smooth", and "Uniform" perform reasonably well. "Repeat" surprisingly performs much worse, likely due to the model confusing the multiple accurate polylines.

V. Conclusion

MapDiffusion is a novel approach that leverages generative diffusion for online vectorized HD map construction in autonomous driving. By integrating a diffusion-based denoising decoder with a learned BEV encoder, MapDiffusion predicts multiple plausible map representations from noisy initial queries. This sampling of the map distribution can also provide spatial uncertainty estimates. Experiments on the nuScenes dataset demonstrated that MapDiffusion achieves state-of-the-art performance, with a relative improvement of 5 % over the StreamMapNet baseline, even without access to queries that are learned or temporally aggregated from the previous frame. Additionally, by sampling outputs, our approach enhances prediction accuracy and generates useful uncertainty maps. Ablation studies revealed optimal configurations for diffusion parameters, query padding strategies, and the impact of pretraining the BEV encoder. Moreover, we showed that the uncertainty maps generated by MapDiffusion estimate significantly higher uncertainty in invisible areas, highlighting their practical relevance for real-world applications. In conclusion, MapDiffusion establishes the new state of the art for online map construction on nuScenes and emphasizes the potential of generative diffusion models in online mapping tasks, proving their ability to enhance accuracy, reliability, and robustness. This generic framework can be applied to other models, improving their performance while providing valuable uncertainty estimates, paving the way for safer and more robust autonomous driving systems.

REFERENCES

[1] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Vectormapnet: End-to-end vectorized hd map learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 22 352–22 369. 1, 2, 5

[2] B. Liao, S. Chen, Y. Zhang, B. Jiang, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Maptrv2: An end-to-end framework for online vectorized hd map construction," *International Journal of Computer Vision*, 2025. 1, 5

[3] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 525–11 533. 1

[4] J. Philion and S. 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. 1, 2

[5] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*. Springer, 2022, pp. 1–18. 1, 2

[6] T. Monninger, V. Dokkadi, M. Z. Anwar, and S. Staab, "TempBEV: Improving learned bev encoders with combined image and bev space temporal aggregation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2024, pp. 9668–9675. 1, 2

[7] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, and X. Zhang, "Petrv2: A unified framework for 3d perception from multi-camera images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3262–3272. 1, 2

[8] T. Yuan, Y. Liu, Y. Wang, Y. Wang, and H. Zhao, "Streammapnet: Streaming mapping network for vectorized online hd map construction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 7356–7365. 1, 2, 3, 5

[9] P. Jia, T. Wen, Z. Luo, M. Yang, K. Jiang, Z. Liu, X. Tang, Z. Lei, L. Cui, B. Zhang, K. Sheng, and D. Yang, "Diffmap: Enhancing map segmentation with map prior using diffusion model," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9836–9843, 2024. 1, 2

[10] D.-T. Le, H. Shi, J. Cai, and H. Rezatofighi, "Diffusion model for robust multi-sensor fusion in 3d object detection and bev segmentation," in *ECCV*. Springer, 2024, pp. 232–249. 1, 2

[11] J. Chen, R. Deng, and Y. Furukawa, "Polydiffuse: Polygonal shape reconstruction via guided set diffusion models," *Advances in neural information processing systems*, vol. 36, 2023. 1, 2, 5

[12] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "Maptr: Structured modeling and learning for online vectorized hd map construction," in *International Conference on Learning Representations*, 2023. 2, 5

[13] T. Monninger, M. Z. Anwar, S. Antol, S. Staab, and S. Ding, "Augmapnet: Improving spatial latent structure via bev grid augmentation for enhanced vectorized online hd map construction," *arXiv preprint arXiv:2503.13430*, 2025. 2

[14] S. Wang, F. Jia, W. Mao, Y. Liu, Y. Zhao, Z. Chen, T. Wang, C. Zhang, X. Zhang, and F. Zhao, "Stream query denoising for vectorized hd-map construction," in *ECCV*. Springer, 2024, pp. 203–220. 2, 5

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, 2020. 2

[16] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019. 2

[17] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, 2021. 2

[18] S. Chen, P. Sun, Y. Song, and P. Luo, "DiffusionDet: Diffusion model for object detection," in *ICCV*, October 2023, pp. 19 830–19 843. 2

[19] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021. 2

[20] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 2

[21] W. Zheng, W. Chen, Y. Huang, B. Zhang, Y. Duan, and J. Lu, "Occworld: Learning a 3d occupancy world model for autonomous driving," in *ECCV*. Springer, 2024, pp. 55–72. 2

[22] G. Wang, Z. Wang, P. Tang, J. Zheng, X. Ren, B. Feng, and C. Ma, "Occgen: Generative multi-modal 3d occupancy prediction for autonomous driving," in *ECCV*. Springer, 2024, pp. 95–112. 2

[23] B. Li, J. Guo, H. Liu, Y. Zou, Y. Ding, X. Chen, H. Zhu, F. Tan, C. Zhang, T. Wang *et al.*, "Uniscene: Unified occupancy-centric driving scene generation," *arXiv preprint arXiv:2412.05435*, 2024. 2

[24] A. Reed, L. Achey, B. Crowe, B. Hayes, and C. Heckman, "Online diffusion-based 3d occupancy prediction at the frontier with probabilistic map reconciliation," *arXiv preprint arXiv:2409.10681*, 2024. 2

[25] X. Gu, M. Zhang, J. Lyu, and Q. Ge, "Generating urban road networks with conditional diffusion models," *ISPRS International Journal of Geo-Information*, vol. 13, no. 6, p. 203, 2024. 2

[26] A. Ruiz, A. Melnik, D. Wang, and H. Ritter, "Lane segmentation refinement with diffusion models," *arXiv preprint arXiv:2405.00620*, 2024. 2

[27] X. Wang, K. Tao, N. Cheng, Z. Yin, Z. Li, Y. Zhang, and X. Shen, "Radiodiff: An effective generative diffusion model for sampling-free dynamic radio map construction," *IEEE Transactions on Cognitive Communications and Networking*, 2024. 2

[28] J. Zou, K. Tian, Z. Zhu, Y. Ye, and X. Wang, "Diffbev: Conditional diffusion model for bird's eye view perception," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7846–7854. 2

[29] M. A. Shabani, S. Hosseini, and Y. Furukawa, "Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5466–5475. 2

[30] X. Gu, G. Song, I. Gilitschenski, M. Pavone, and B. Ivanovic, "Producing and leveraging online map uncertainty in trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 521–14 530. 2

[31] X. Han, H. Zheng, and M. Zhou, "Card: Classification and regression diffusion models," in *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. 2

[32] Z. Du and J. Li, "Diffusion-based probabilistic uncertainty estimation for active domain adaptation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 17 129–17 155, 2023. 2

[33] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov *et al.*, "Motiondiffuser: Controllable multi-agent motion prediction using diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9644–9653. 2

[34] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*. Springer, 2020, pp. 213–229. 3

[35] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021. 3

[36] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021. 4, 6

[37] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 5

[38] G. Zhang, J. Lin, S. Wu, Z. Luo, Y. Xue, S. Lu, Z. Wang *et al.*, "Online map vectorization for autonomous driving: A rasterization perspective," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 5

[39] X. Liu, S. Wang, W. Li, R. Yang, J. Chen, and J. Zhu, "Mgmap: Mask-guided learning for online vectorized hd map construction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 812–14 821. 5

[40] Z. Liu, X. Zhang, G. Liu, J. Zhao, and N. Xu, "Leveraging enhanced queries of point sets for vectorized map construction," in *ECCV*, 2024. 5

[41] Y. Zhou, H. Zhang, J. Yu, Y. Yang, S. Jung, S.-I. Park, and B. Yoo, "Himap: Hybrid representation learning for end-to-end vectorized hd map construction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 5

[42] X. Tian, T. Jiang, L. Yun, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," *Advances in neural information processing systems*, vol. 36, 2023. 5