

# 3D PERCEPTION WITH DIFFERENTIABLE MAP PRIORS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Human drivers rarely travel where no person has gone before. After all, thousands of drivers use busy city roads every day, and only one can claim to be the first. The same holds for autonomous computer vision systems. The vast majority of the deployment area of an autonomous vision system will have been visited before. Yet, most autonomous vehicle vision systems act as if they are encountering each location for the first time. In this work, we present Differentiable Map Priors (**DMP**), a simple but effective framework to learn spatial priors from historic traversals. Differentiable Map Priors easily integrate into leading 3D perception systems at little to no extra computational costs. We show that they lead to a significant and consistent improvement in 3D object detection and semantic map segmentation tasks on the nuScenes dataset across several architectures.

## 1 INTRODUCTION

Autonomous vehicles rarely visit a truly unseen location. Current deployments are typically geofenced to operate within a known, carefully mapped area. Later, fleet deployments will cover the same area over and over again, collecting massive amounts of rich sensor data from the same locations. Yet, current perception systems mostly treat the static world as never been seen before, and jointly infer both static and dynamic scene structures from sensor inputs alone Liu et al. (2022d); Peng et al. (2023); Wang et al. (2023a); Liu et al. (2022c); Huang & Huang (2022).

In this work, we equip vision models with a persistent memory of the world. We build up this memory as part of the training of the perception system in a Differentiable Map Prior (**DMP**). Our map prior is trained end-to-end for the downstream task, allowing a 3D perception system to utilize its knowledge of location and past experiences to enhance its predictions. At test time, the perception system leverages the learned map, which is enriched with a wealth of features built up during training. This comprehensive prior knowledge serves to augment the capabilities of the underlying perception stack, allowing the system to make more informed and accurate inferences about the surrounding environment. We design our map with a compact and memory-efficient representation to ensure scalability for real-world applications.

To validate the efficacy of our approach, we conduct extensive experiments on the nuScenes Caesar et al. (2020) dataset and incorporate DMP into three distinct multi-view perception stacks. These baselines include both transformer-based Liu et al. (2022b) and convolutional Huang et al. (2021); Li et al. (2022b) perception systems. Our experiments show that incorporating our Differentiable Map Prior yields consistent performance gains across all evaluated baselines.

## 2 RELATED WORKS

**Camera-based 3D Perception.** Camera-only perception systems are a compelling choice for autonomous vehicles due to their high resolution and cost-effectiveness. While many highly accurate perception systems focused on monocular 3D object detection Wang et al. (2021); Zhou et al. (2019); Park et al. (2021), modern autonomous vehicles utilize multiple cameras with potentially overlapping fields of view. To this extent, an increasing number of research efforts have shifted towards multi-view approaches Liu et al. (2022b); Li et al. (2022b); Xiong et al. (2023a); Huang et al. (2021); Yang et al. (2023b); Zhou & Krähenbühl (2022) which enable perception systems to construct a more comprehensive internal representation of the environment.

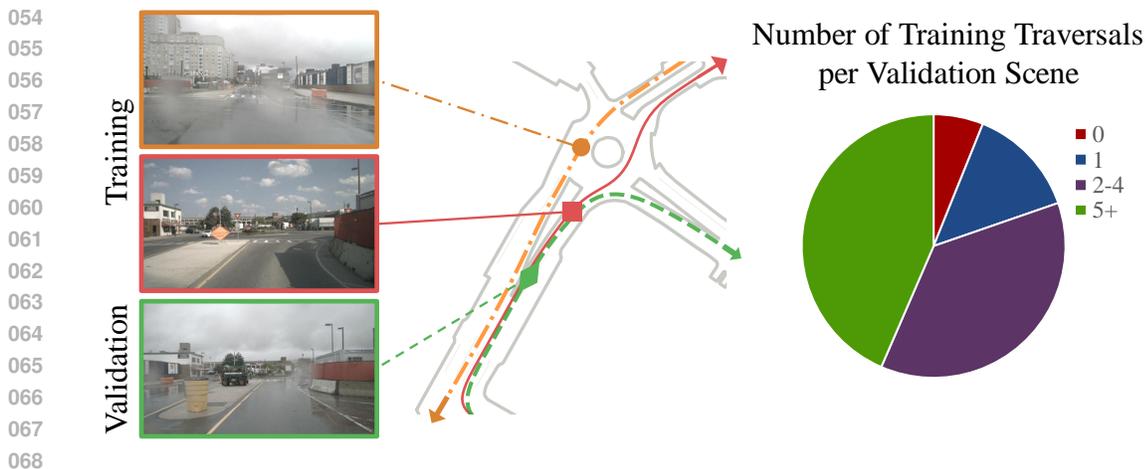


Figure 1: **Visualization of overlapping routes in the nuScenes benchmark.** Left: A visual example of different traversals of the same scene in nuScenes. Each captures the same intersection from a slightly different vantage point. Center: A visualization of the routes driven. Right: The fraction of validation scenes with no overlap (0) with the training set, or with an overlap with  $n > 0$  training routes. The vast majority of validation scenes heavily overlap with training scenes.

One line of work Phillion & Fidler (2020); Li et al. (2022a); Reading et al. (2021) aggregates image features to a canonical “BEV frame” by predicting dense categorical depth for each image and pooling image features from a virtual frustum. Alternatively, BEV representations can be built using attention across camera views with geometric positional embeddings Xiong et al. (2023a); Chen et al. (2022a); Zhou & Krähenbühl (2022). Another approach Liu et al. (2022b); Wang et al. (2022) bypasses the explicit BEV representation, directly performing attention across the multi-view images in a DETR Carion et al. (2020) fashion.

These models have been applied to a variety of tasks, showcasing their versatility and effectiveness in understanding the surrounding environment. Object detection Liu et al. (2022b); Li et al. (2022a); Wang et al. (2022); Chen et al. (2022b) has been a primary focus, serving as a key role for autonomous vehicles. In addition, these models have been applied to HD-Map creation via semantic map prediction Phillion & Fidler (2020); Li et al. (2022b); Zhou & Krähenbühl (2022); Liu et al. (2022d); Hu et al. (2021), and vectorized map prediction Li et al. (2021); Liu et al. (2022a); Liao et al. (2022). These methods assume that each scene is encountered for the first time, overlooking valuable information from prior traversals. Our proposed DMP augments these models by integrating a persistent view of static scene elements from past traversals into the perception pipeline.

**Perception with Historical Context.** Recent works Saha et al. (2021); Wang et al. (2023b); Yang et al. (2022); Huang & Huang (2022) have developed models that incorporate temporal context, demonstrating improvements over their single-frame counterparts. While these approaches focus on modeling *temporal information*, our work focuses on the *spatial aspect*, leveraging the fact that the same area is traversed multiple times.

For LiDAR-based detection, Hindsight You et al. (2022) precomputes and stores quantized features from historical point cloud data. At test time, they augment the current scene with geo-indexed historical features, resulting in an improved detection performance. This closely resembles our differentiable map prior: Features computed at training time, help inference at test time. The key difference is the representation of the prior. Hindsight uses an explicit point-based prior, while DMP uses a much more compact implicit function-based representation that is learned jointly with the perception system during training.

A recent line of work has explored incorporating historical data into camera-based perception systems. NMP Xiong et al. (2023b) targets static semantic map segmentation from multi-view camera inputs by augmenting live sensor features with historical features using a GRU-based fusion module. They recursively update their prior by directly storing the features into the global map, represented as a set of dense “tiles”, and show this external memory improves segmentation performance. In

108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161

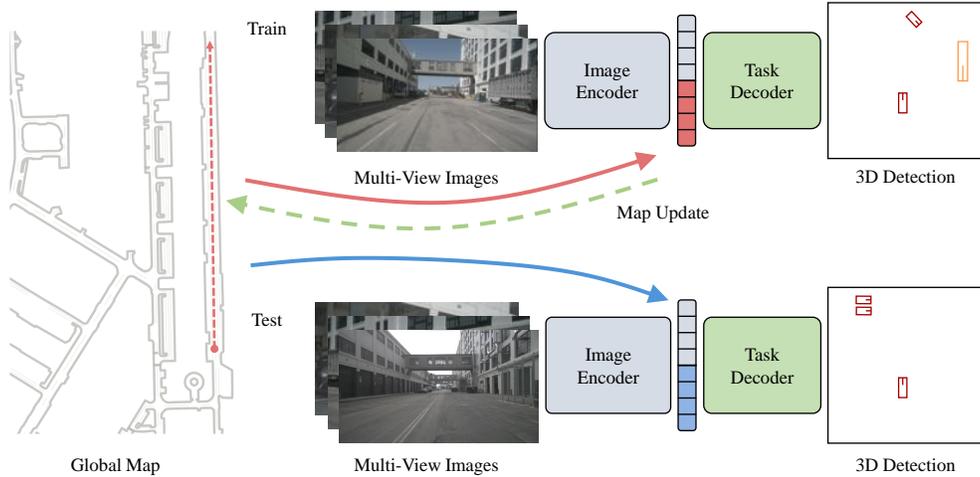


Figure 2: **Overview of Differentiable Map Priors (DMP)**. The framework employs a global map that implicitly encodes useful information as latent vectors for future traversal. During training, the latent vector of a particular location is used in combination with the sensory feature. The maps are modeled with learnable parameters and are thus fully differentiable. During testing, we leverage the learned map priors to enhance the features for downstream tasks.

contrast, PreSight Yuan et al. (2024) utilizes historical data to model entire cities as a collection of neural fields Müller et al. (2022); Yang et al. (2023a). They optimize for this representation using a photometric reconstruction loss and at test time, they voxelize these reconstructed features and incorporate them into online perception models. While these methods showcase the benefits of incorporating historical data into camera-based vision models, they rely on learning and storing these features in a procedural manner with substantial memory requirements. DMP learn the map features in an end-to-end differentiable manner as part of the overall training pipeline of the perception system. This allows the model to implicitly learn what features are useful for the downstream task, and how to store them in the most compact implicit representation.

### 3 PRELIMINARIES

**Multi-view 3D detectors** ingest  $n$  camera images  $I_1, \dots, I_n$  with  $I_k \in \mathbb{R}^{H \times W \times 3}$  and corresponding 2D pose information  $p_k$  to 3D bounding boxes  $B = \{b_1, b_2, \dots, b_m\}$ , where each bounding box  $b_i$  is represented by its center, dimensions, orientation, and class score. Internally, the model extracts feature  $\mathbf{X}_i$  from each image, then builds an intermediate representation  $\mathbf{X}_{sensor}$  using an encoder  $E$  and finally decodes each intermediate representation into a potential detection with a decoder  $D$ . 3D detectors fall in two broad categories: Dense map-based detectors Huang et al. (2021); Li et al. (2022b); Yang et al. (2022); Liu et al. (2022d) and transformer based detectors Wang et al. (2022); Liu et al. (2022b;c).

A dense map-based detector encodes a feature map  $\mathbf{X}_{sensor} \in \mathbb{R}^{h_{bev} \times w_{bev} \times c}$  that represents a local map-region of size  $h_{bev} \times w_{bev}$  around the vehicle. The decoder operates directly on this intermediate map representation and translates each spatial location into a potential detection with an associated score. Transformer-based detectors use a much sparser intermediate representation. They start from a set of  $m$  learned queries  $Q = \{q_1, \dots, q_m\}$  which cross-attend to image features  $\mathbf{X}_i$ . The resulting sparse feature representation  $\mathbf{X}_{sensor} \in \mathbb{R}^{m \times c}$  forms the input to a transformer-based decoder. BEVFormer Li et al. (2022b) uses concepts from both: a map-based BEV representation forms queries for a transformer-based multi-view encoder-decoder. Our differentiable map prior applies to both kinds of architecture, albeit with a slightly different fusion architecture.

**Maps.** The simplest maps are dense 2D grids, where each cell contains a  $d$ -dimensional feature. Dense representations quickly become memory-bound and are less suitable for large-scale applications. Similar challenges have been observed for neural scene representations of complete 3D scenes Müller et al. (2022); Yu et al. (2022). For 3D scenes, sparse representations Müller et al.

162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215

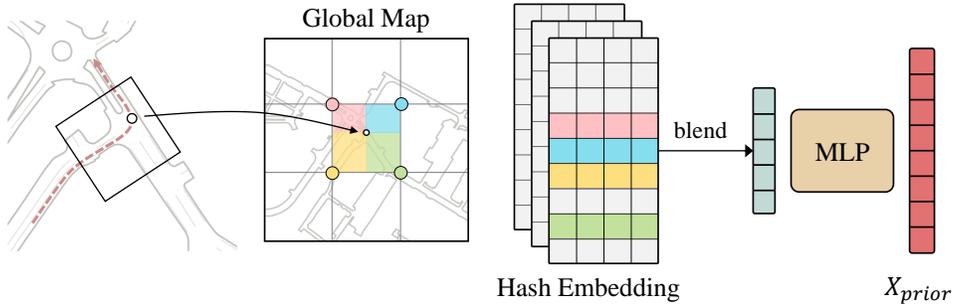


Figure 3: **Prior map representation.** We start from a regularly sampled grid of locations around the vehicle. For each grid point, we sample its four nearest grid points and bilinearly interpolate their corresponding embeddings. We repeat this process at multiple levels of resolutions and concatenate the embeddings across levels. Finally, we use an MLP to project the multi-level feature into a fixed-sized prior feature.

(2022); Yu et al. (2022) have gained popularity due to their memory efficiency, with one of the most popular being multi-resolution hash embeddings Müller et al. (2022). These hash embeddings map an input  $\mathbf{x} \in \mathbb{R}^D$  to a series of  $L$  hash maps at different resolutions, where each hash map contains a learned embedding of size  $T$ . The result is a feature  $m(\mathbf{x}) \in \mathbb{R}^d$  that concatenates hash lookups across all hash levels. Finally, an MLP projects these hashed feature values into a representation used by a neural renderer. Our differentiable map priors directly build on multi-resolution hash maps as their underlying representation. Specifically, we use a sparse 2D multi-resolution hash map  $m : \mathbb{R}^2 \rightarrow \mathbb{R}^d$  to store spatial map-view feature representations.

#### 4 DIFFERENTIABLE MAP PRIORS

Our differentiable map prior has two components: A sparse and differentiable map representation builds up a feature representation  $\mathbf{X}_{prior}$  of the static scene. A fusion module then splices spatial map features into an existing 3D detection architecture. See Figure 2 for an overview.

At training time, we differentiate through the map representation to learn a persistent static map representation that helps the 3D detector improve its accuracy. At inference time, we simply augment the detector with a spatial map prior when available, or fall back to a map-less model in novel areas.

**Map Representation.** Our global map representation  $\mathbf{X}_{prior} \in \mathbb{R}^{h \times w \times d}$  directly builds on a multi-resolution hash map  $m : \mathbb{R}^2 \rightarrow \mathbb{R}^{\hat{d}}$  Müller et al. (2022). For each location  $\mathbf{x} \in \mathbb{R}^2$  the hash map returns a potentially interpolated feature  $m(\mathbf{x}) \in \mathbb{R}^{\hat{d}}$ . To retrieve the prior features  $\mathbf{X}_{prior}$  we start from the vehicle pose  $p$ . We discretize a  $w \times h$  region of interest around the vehicle into a grid  $g^{local} \in \mathbb{R}^{h \times w \times 2}$ , where each cell  $g_{i,j}^{local}$  corresponds to a point in the coordinate frame relative to the ego-vehicle. The prior features then sample and inflate the multi-resolution hash map representation

$$\mathbf{X}_{prior}(r) = \text{MLP}(m(M_p g^{local})),$$

where  $M_p$  is the affine transformation corresponding to the vehicle pose  $p$ . The multi-layer perceptron MLP allows the hash map to use a much smaller dimensionality  $\hat{d} \ll d$  than the actual prior  $\mathbf{X}_{prior}$ . This saves a significant amount of memory and allows for large scale city-wide map representations. Figure 3 shows an overview of the process. We implement the multiresolution hash embedding in pure PyTorch, resulting in a map query time with negligible computational overhead (less than 2% of the overall detector’s runtime).

**Prior Fusion.** We use a fusion module  $F$  to incorporate the map prior features  $\mathbf{X}_{prior}$  into the onboard sensor features  $\mathbf{X}_{sensor}$ . The onboard sensor features  $\mathbf{X}_{sensor}$  are extracted from the multi-view camera images and are used as input to the 3D detector.

For dense map-based detectors Huang et al. (2021), including BEVFormer Li et al. (2022b), we align the size of the map representation  $\mathbf{X}_{prior}$  to the size of the intermediate birds-eye-view sensor

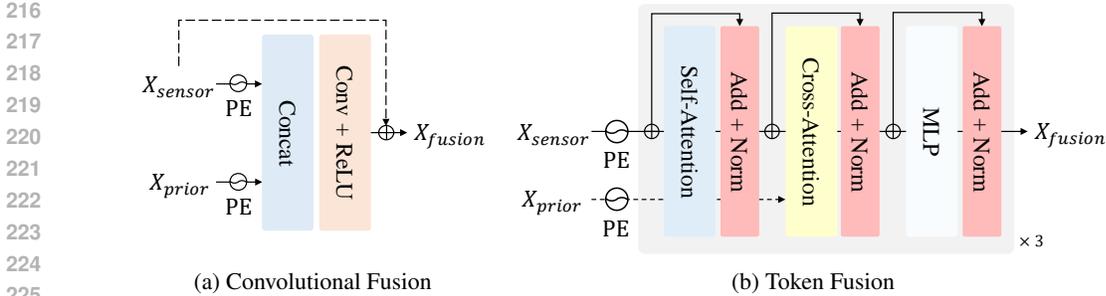


Figure 4: **Prior fusion.** Convolutional Fusion (a) concatenates the sensor feature and the map prior feature and fuses them with a residual module. Token Fusion (b) uses the map prior feature to query the sensor feature with a cross-attention module. Both sensor features and map prior features are modulated with positional embeddings before the fusion operation.

representation  $\mathbf{X}_{sensor}$ . We then concatenate the two and fuse them using a single  $3 \times 3$  convolution with ReLU. We make sure the number of output channels if  $\mathbf{X}_{fusion}$  matches the number of sensor channels  $\mathbf{X}_{sensor}$  for seamless integration into existing detection architectures. We experimentally found that adding a positional embedding to both the map prior  $\mathbf{X}_{prior}$  and sensor representation  $\mathbf{X}_{sensor}$  improved the model’s performance. We apply this fusion step exactly once onto the final sensor features.

For transformer-based detectors Wang et al. (2022); Liu et al. (2022b;c), we fuse the map-prior into the sparse sensor embeddings  $\mathbf{X}_{sensor}$  for each query. We use a cross-attention layer to join the sensor and map prior features. Since the sparse sensor features  $\mathbf{X}_{sensor}$  do not align with our map prior  $\mathbf{X}_{prior}$ , we use a positional embedding for both sensor and prior features. For prior features, we learn the positional embedding as a set of free parameters  $\mathbf{E}_{prior} \in \mathbb{R}^{x \times h \times d}$ . For sensor embeddings, we use a linear projection of the learned positional embedding of the query in the original detector. Figure 4b shows an overview of the transformer-based fusion process. We again apply this fusion step exactly once in one of the last transformer blocks.

Despite its simplicity our differentiable map prior yields a significant improvement in performance across all baseline architectures.

## 5 EXPERIMENTS

We evaluate the efficacy of adding historical priors with DMP for 3D object detection from multi-view camera images across a variety of architectures.

### 5.1 DATASET

We conduct experiments on the nuScenes Caesar et al. (2020) dataset, which consists of 850 scenes (700 training, 150 validation), covering two cities - Boston and Singapore. Each scene is 20 seconds long, with 3D bounding box annotations at 2 Hz for a total of 40 frames per scene. For each frame, we use the six multi-view camera images with their calibrated intrinsics and pose information and ego-pose. A large proportion of the scenes in the dataset have been traversed multiple times, as shown in Figure 5. For these, our map prior natively applies. For scenes without any overlap with training, we fall back onto the baseline algorithm.

**Metrics.** Aligning with the standard 3D detection evaluation methodology, we report mean Average Precision (**mAP**) across all 10 classes, calculated using ground plane center distance for matching predicted and ground truth results. Additional metrics include five true positive metrics (ATE, ASE, AOE, AVE, AAE) for measuring errors in translation, scale, orientation, velocity, and attributes. The nuScenes Detection Score (**NDS**) Caesar et al. (2020) is a weighted sum of the mAP and the true positive metrics and provides a comprehensive evaluation of a model’s performance.

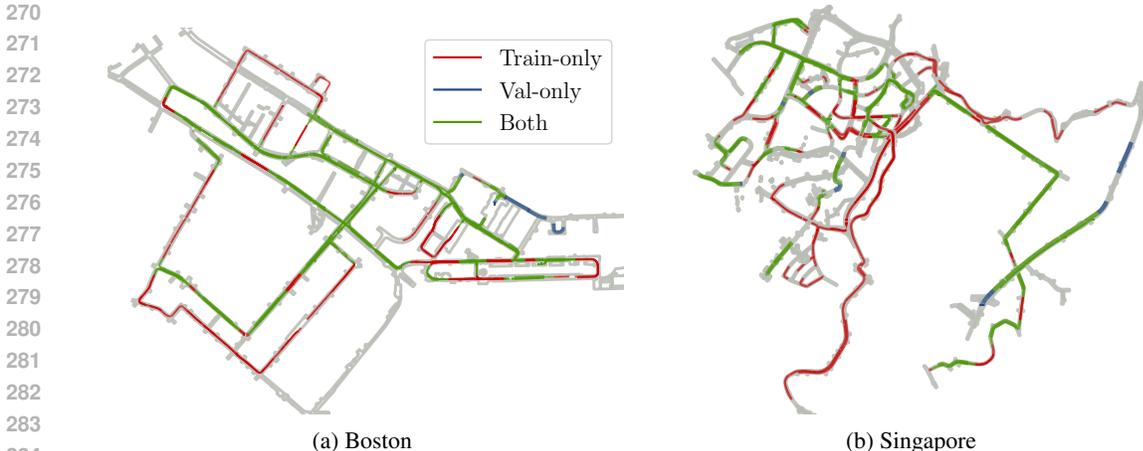


Figure 5: **Map visualization of the nuScenes Caesar et al. (2020) dataset.** We delineate the traversals from the training and validation split of the dataset in bold colors. “Both” denotes scenes that have been traversed in both the training and validation splits. “Val-only” refers to scenes that have no significant overlap (within 50m) with any training scenes and are geographically disjoint from the training/validation set. “Train-only” refers to scenes that have no significant overlap with any validation scenes.

## 5.2 IMPLEMENTATION DETAILS

We apply our method to three different 3D object detection architectures: BEVDet Huang et al. (2021), BEVFormer Li et al. (2022b), and PETR Liu et al. (2022b). These models represent the best-performing models for 3D object detection and cover the two most used architectural paradigms: dense BEV architectures and end-to-end transformer architectures. For each baseline, we use the single-timestamp model variant with only multi-view camera inputs and pose information for a fair comparison.

For our prior storage, as described in Section 4, we use a multi-level hash embedding with  $L = 4$  levels each with  $T = 2^{16}$  learned embedding of size 8 for a total of 32 dimensions. The finest resolution has a size of 0.5 meters per voxel, and the coarsest resolution has a size of 25 meters per voxel. The MLP consists of 3 layers and projects the retrieved embeddings to  $\mathbf{X}_{prior} \in \mathbb{R}^{w \times h \times 128}$ , where the width  $w$  and height  $h$  match the baseline algorithms internal feature resolution. BEVDet Huang et al. (2021) has a resolution  $w = h = 128$  and BEVFormer Li et al. (2022b) uses a resolution of  $w = h = 150$ . For PETR Liu et al. (2022b), we use a coarser resolution  $w = h = 64$ .

Both the BEVDet and BEVFormer models use a ResNet-101 image backbone initialized with weights from a pre-trained FCOS3D Wang et al. (2021), and PETR uses a VoVnet-99 Lee et al. (2019) initialized from a DD3D Park et al. (2021) checkpoint. In BEVDet and BEVFormer, we use the same BEV augmentations (flipping, scaling, rotating) and apply the same augmentation accordingly when retrieving the prior features.

Table 1: **nuScenes 3D object detection** with Differential Map Priors, reported on the official validation split. Incorporating learned priors with DMP improves performance across all baselines.

Method	DMP	NDS $\uparrow$	mAP $\uparrow$	mASE $\downarrow$	mAAE $\downarrow$	mAVE $\downarrow$	mAOE $\downarrow$	mATE $\downarrow$
BEVDet Huang et al. (2021)	✓	0.338	0.262	0.299	0.238	0.860	0.758	0.776
		<b>0.381</b>	<b>0.302</b>	<b>0.301</b>	<b>0.234</b>	<b>0.751</b>	<b>0.786</b>	<b>0.629</b>
PETR Liu et al. (2022b)	✓	0.403	0.339	0.279	0.182	0.931	0.531	0.793
		<b>0.422</b>	<b>0.349</b>	<b>0.277</b>	<b>0.168</b>	<b>0.836</b>	<b>0.530</b>	<b>0.766</b>
BEVFormer Li et al. (2022b)	✓	0.419	0.320	0.279	<b>0.145</b>	0.763	0.448	0.776
		<b>0.438</b>	<b>0.348</b>	0.280	0.164	0.763	<b>0.439</b>	0.678

Across all models, we train for 24 epochs using the AdamW Loshchilov & Hutter (2019) optimizer using a learning rate  $2 \times 10^{-4}$  with a cosine annealing schedule. All experiments are performed on a single-node machine with 8 Titan-V GPUs and a total batch size of 8. The full training duration is approximately 1 day.

### 5.3 MAIN RESULTS

Shown in Table 1, we compare the performance of adding DMP across BEVFormer Li et al. (2022b), BEVDet Huang et al. (2021) and PETR Liu et al. (2022b) on the nuScenes validation set. We use the predictions from the corresponding baseline without the prior for any location that has not been traversed during training. Across all baselines, we observe consistent improvements across all evaluation metrics.

We observe that the two BEV-based architectures, BEVFormer and BEVDet, benefit the most from the addition of the prior, with BEVDet showing the largest improvement (relative 13% NDS and 15% mAP). The architectures with explicit spatial BEV representations are likely to benefit more from the prior as the prior features are well aligned with the model’s internal representation. In contrast, the fully transformer architecture (PETR) has to perform additional spatial reasoning to connect the prior features with its detection queries.

**Comparison with prior work.** We compare DMP to NMP Xiong et al. (2023b), which similarly enhances online map perception with learned priors. NMP uses location information to retrieve a local BEV feature map from external dense storage and fuse it with online sensor features. In their framework, they build separate priors for training/testing. To closer match our setting, where we are interested in improving performance in areas previously traveled, we modify NMP and use the training prior during evaluation.

We apply the adapted method, denoted as NMP\*, for 3D object detection and train DMP on for semantic map prediction. The map segmentation task consists of three classes: divider, pedestrian crossing, and road boundaries, and we report mean Intersection over Union (mIoU) across these classes. For this comparison, we use BEVFormer with a slightly smaller spatial resolution of  $w = h = 100$  as the base architecture and use the original detection head along with an extra head for predicting map segmentation. We train the model jointly with the original detection loss and a weighted cross-entropy loss for segmentation.

Table 2: **Comparison with NMP** Xiong et al. (2023b) on joint object detection and map segmentation. NMP\* denotes that the prior learned during training is used during evaluation.

Prior	NDS↑	mAP↑	mIoU↑
-	0.334	0.258	0.217
NMP*	0.347	0.273	0.297
DMP	<b>0.368</b>	<b>0.284</b>	<b>0.568</b>

We show the comparison with their alternative prior model in Table 2. While both learned priors show improvements over the baseline in both tasks, our method achieves a more significant improvement in both detection and segmentation. For the map prediction task, all the segmentation classes (divider, pedestrian crossing, road boundaries) are static, and our method can trivially learn to embed the map into the prior features. Moreover, using DMP shows a larger improvement in detection performance, demonstrating the effectiveness of our method as a prior for autonomous driving perception.

We hypothesize this performance boost for detection is a result of the end-to-end learned prior features, allowing the model to learn what features are useful for the downstream task. In contrast, NMP captures prior features in a non-differentiable manner.

### 5.4 ABLATIONS

**Performance with Multiple Traversals.** We study how the number of traversals seen during training affects model performance in Figure 6. A traversal is defined as each sample being  $<50\text{m}$  from any other sample seen during training. We split the dataset into 3 subsets based on the number of

378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431

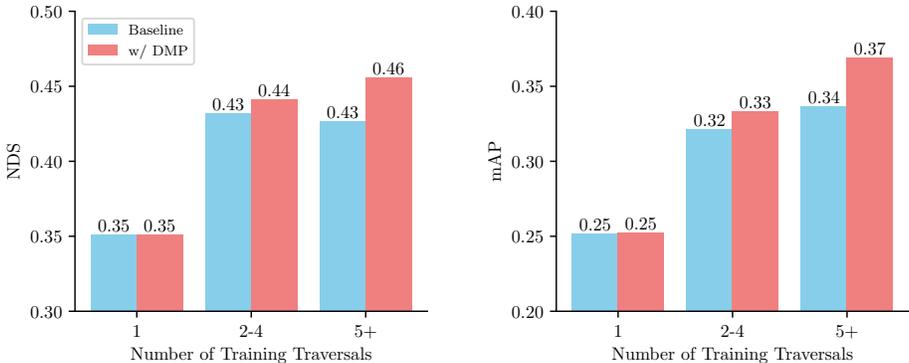


Figure 6: **Performance with different numbers of training traversals.** Both the baseline and our method do much better as the number of traversals increases, suggesting that even a baseline learns to recognize its training environment and a potential static prior. An explicit prior in the form of DMP performs significantly better.

traversals: (1) a single traversal, (2) 2-4 traversals, and (3) 5+ traversals. We can see that DMP consistently improves the baseline without the map prior when more than one traversals are given. The gain magnifies as the number of traversals increases, demonstrating the effectiveness of explicitly modeling map priors in 3D perception.

**Map Embedding Size.** As described in Section 4, the historical feature storage utilizes a multi-level hash embedding with several hyperparameters. In Figure 7, we ablate the impact of varying the sparsity level of the underlying map storage on the overall performance of our method. We adjust the number of embeddings per hash level  $T$ , affecting the granularity of the learned spatial representation with respect to the resolution. In this setting, we keep the rest of the encoder hyperparameters fixed, as specified in Section 5.2

Increasing the embedding size enhances the expressiveness of prior features, leading to better detection accuracy. However, this comes at the cost of increased memory requirements for storing the embeddings and experimental results show an embedding size  $T = 2^{16}$  balances performance and memory, with diminishing returns for larger embedding sizes.

**Model Latency.** Table 3 shows the computation overhead of our method with respect to the baseline model’s total latency. We provide timing results over 100 samples, measured on a single Titan-V GPU using a batch size of 1. Incorporating our prior incurs a relatively small overhead, about  $\sim 3\%$  of the full model’s latency.

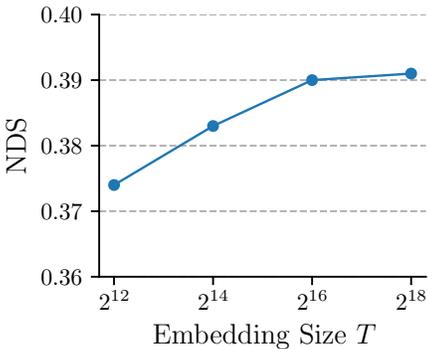


Figure 7: Resolution of the underlying hash table in our learned map representation.

Table 3: **Computational Overhead.** Execution time of the prior sampling.

Operation	Time (ms)	% Total
Prior Sampling	7.63 ± 0.09	2.50%
Prior Fusion	1.57 ± 0.03	0.51%
Forward Pass	305.17 ± 0.37	–

**Distance Falloff.** We conduct an analysis across three distinct distance thresholds: “close” (0-10 meters), “medium” (10-25 meters), and “far” (25-50 meters). We measure the detection precision of two representative classes: “car” and “barrier” in Figure 8.

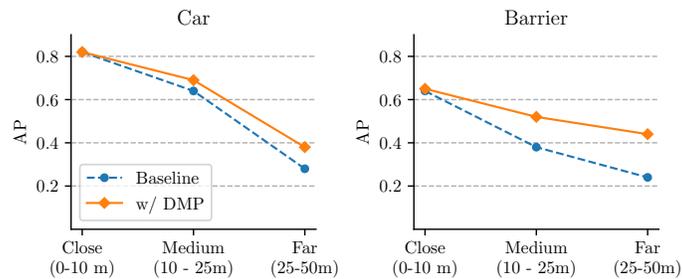


Figure 8: **Performance over different distances.** We compare the performance of DMP across three different distance thresholds: close (0-10 meters), medium (10-25 meters), and far (25-50 meters).

Our approach demonstrates a more graceful degradation in performance with respect to distance, suggesting that the prior aids in the detection of objects located farther away. Barriers are inherently static objects, it should thus not come as a surprise that the incorporation of prior knowledge about their location and geometry from previous traversals allows models equipped with our priors to achieve significantly higher detection accuracy.



Figure 9: **Qualitative Results.** Predictions from BEVFormer+DMP on the nuScenes validation set.

## 6 CONCLUSION

We present a new framework for incorporating historical context into perception models with Differentiable Map Priors (DMP). We evaluate our method on the nuScenes dataset and show consistent improvements across a variety of architectures, showing that it is indeed possible to leverage previous traversals for detection from multi-view images. Our framework is simple and effective and designed with scalability in mind for real-world applications.

## REFERENCES

- 486  
487  
488 Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush  
489 Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for  
490 autonomous driving. In *CVPR*, pp. 11621–11631, 2020.
- 491 Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey  
492 Zagoruyko. End-to-end object detection with transformers. In *European conference on computer  
493 vision*, pp. 213–229. Springer, 2020.
- 494 Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu.  
495 Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer.  
496 *arXiv preprint arXiv:2206.04584*, 2022a.
- 497 Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor  
498 fusion framework for 3d detection. *arXiv preprint arXiv:2203.10642*, 2022b.
- 500 Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto  
501 Cipolla, and Alex Kendall. Fiery: future instance prediction in bird’s-eye view from surround  
502 monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer  
503 Vision*, pp. 15273–15282, 2021.
- 504 Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward  
505 deployment. *arXiv preprint arXiv:2211.17111*, 2022.
- 507 Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance  
508 multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.
- 509 Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and  
510 gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the  
511 IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.
- 512 Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: A local semantic map learning and  
513 evaluation framework. *arXiv preprint arXiv:2107.06307*, 2021.
- 515 Yin hao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and  
516 Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv  
517 preprint arXiv:2206.10092*, 2022a.
- 518 Zhiqi Li, Wenhao Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai.  
519 Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal  
520 transformers. *arXiv preprint arXiv:2203.17270*, 2022b.
- 522 Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and  
523 Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction.  
524 *arXiv preprint arXiv:2208.14437*, 2022.
- 525 Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map  
526 learning. *arXiv preprint arXiv:2206.08920*, 2022a.
- 527 Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation  
528 for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022b.
- 530 Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian  
531 Sun. Petr2: A unified framework for 3d perception from multi-camera images. *arXiv preprint  
532 arXiv:2206.01256*, 2022c.
- 533 Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han.  
534 Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. *arXiv  
535 preprint arXiv:2205.13542*, 2022d.
- 536 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.
- 537 Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics  
538 primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 2022.
- 539

- 540 Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for  
541 monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on*  
542 *Computer Vision*, pp. 3142–3152, 2021.
- 543  
544 Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird’s  
545 eye view semantic segmentation from arbitrary camera rigs. In *Proceedings of the IEEE/CVF*  
546 *Winter Conference on Applications of Computer Vision*, pp. 5935–5943, 2023.
- 547  
548 Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by  
549 implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pp. 194–210. Springer,  
550 2020.
- 551  
552 Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution  
553 network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on*  
*Computer Vision and Pattern Recognition*, pp. 8555–8564, 2021.
- 554  
555 Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Enabling spatio-temporal  
556 aggregation in birds-eye-view vehicle estimation. In *2021 IEEE International Conference on Robotics*  
557 *and Automation (ICRA)*, pp. 5133–5139. IEEE, 2021.
- 558  
559 Haiyang Wang, Hao Tang, Shaoshuai Shi, Aoxue Li, Zhenguo Li, Bernt Schiele, and Liwei Wang.  
560 Unitr: A unified and efficient multi-modal transformer for bird’s-eye-view representation. In  
561 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6792–6802,  
2023a.
- 562  
563 Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric  
564 temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF*  
*International Conference on Computer Vision*, pp. 3621–3631, 2023b.
- 565  
566 Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage  
567 monocular 3d object detection. *arXiv preprint arXiv:2104.10956*, 2021.
- 568  
569 Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin  
570 Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference*  
571 *on Robot Learning*, pp. 180–191. PMLR, 2022.
- 572  
573 Kaixin Xiong, Shi Gong, Xiaoqing Ye, Xiao Tan, Ji Wan, Errui Ding, Jingdong Wang, and Xiang Bai.  
574 Cape: Camera view position embedding for multi-view 3d object detection. In *Proceedings of the*  
*IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21570–21579, 2023a.
- 575  
576 Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Neural map  
577 prior for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision*  
*and Pattern Recognition*, pp. 17535–17544, 2023b.
- 578  
579 Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang,  
580 Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to  
581 bird’s-eye-view recognition via perspective supervision. *arXiv preprint arXiv:2211.10439*, 2022.
- 582  
583 Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei  
584 Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition  
585 via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023a.
- 586  
587 Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li Wang, Xinyu Zhang, and Peng Chen.  
588 Bevheight: A robust framework for vision-based roadside 3d object detection. In *Proceedings of*  
*the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21611–21620, 2023b.
- 589  
590 Yurong You, Katie Z Luo, Xiangyu Chen, Junan Chen, Wei-Lun Chao, Wen Sun, Bharath Hariharan,  
591 Mark Campbell, and Kilian Q Weinberger. Hindsight is 20/20: Leveraging past traversals to aid 3d  
592 perception. *ICLR*, 2022.
- 593  
594 Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo  
Kanazawa. Plenoxels: Radiance fields without neural networks. *CVPR*, 2022.

594 Tianyuan Yuan, Yucheng Mao, Jiawei Yang, Yicheng Liu, Yue Wang, and Hang Zhao. Pre-  
595 sight: Enhancing autonomous vehicle perception with city-scale nerf priors. *arXiv preprint*  
596 *arXiv:2403.09079*, 2024.  
597  
598 Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic  
599 segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*  
600 *recognition*, pp. 13760–13769, 2022.  
601 Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint*  
602 *arXiv:1904.07850*, 2019.  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647