A Additional Analysis

A.1 Hash Embedding Size

The compressed map representation relies on a multi-level spatial hash encoding with hash table size T as a key hyperparameter controlling the trade-off between representational capacity and memory efficiency. We systematically evaluate this trade-off by varying T while keeping all other encoder hyperparameters fixed as specified in Section 5.1.

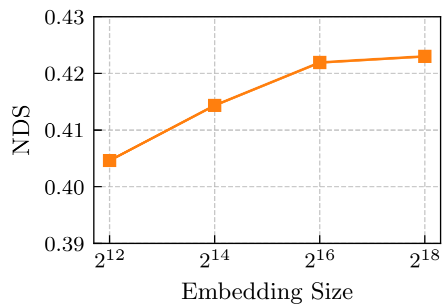


Figure 7: **Embedding Size.** The effect of embedding size T vs. downstream detection performance.

Figure 7 demonstrates that increasing embedding size enhances the expressiveness of prior features, leading to improved detection accuracy. However, this improvement comes at the cost of increased memory requirements for storing the embeddings. Our experimental results reveal that an embedding size of $T = 2^{16}$ achieves an optimal balance between performance and memory efficiency, with diminishing returns observed for larger embedding sizes.

A.2 Comparison with Neural Map Priors

While our approach shares the high-level motivation of leveraging spatial priors with Neural Map Priors (NMP) Xiong et al. (2023b), several fundamental architectural and methodological differences distinguish our work and enable superior performance in the autonomous driving domain.

The most significant distinction lies in our training methodology. NMP employs gradient detaching during feature aggregation, training only the prior fusion module while keeping spatial features frozen. This design choice limits the model's ability to jointly optimize spatial representations with downstream task objectives. In contrast, our method enables end-to-end learning of positional embeddings alongside the detection task, allowing both prior parameters and fusion modules to be jointly optimized through the detection loss. This integrated optimization leads to more effective spatial representations tailored to the specific perception objectives.

Our approach also differs fundamentally in prior persistence and knowledge transfer. NMP constructs new priors during online inference, effectively discarding valuable spatial knowledge accumulated during training. This design is suboptimal for autonomous navigation scenarios where vehicles repeatedly traverse familiar routes. Our method maintains persistent priors that transfer learned spatial knowledge from training to inference, making it particularly well-suited for the predominantly repetitive traversal patterns in autonomous driving.

From a memory efficiency perspective, NMP relies on dense feature tiles that incur significant storage overhead. Our binarized hash-based encoding achieves a 20$\times$ memory reduction (32 KB/km^2 vs 640 KB/km^2) while maintaining comparable detection performance. This efficiency gain is crucial for deployment in resource-constrained autonomous vehicle platforms.

B Experimental Details

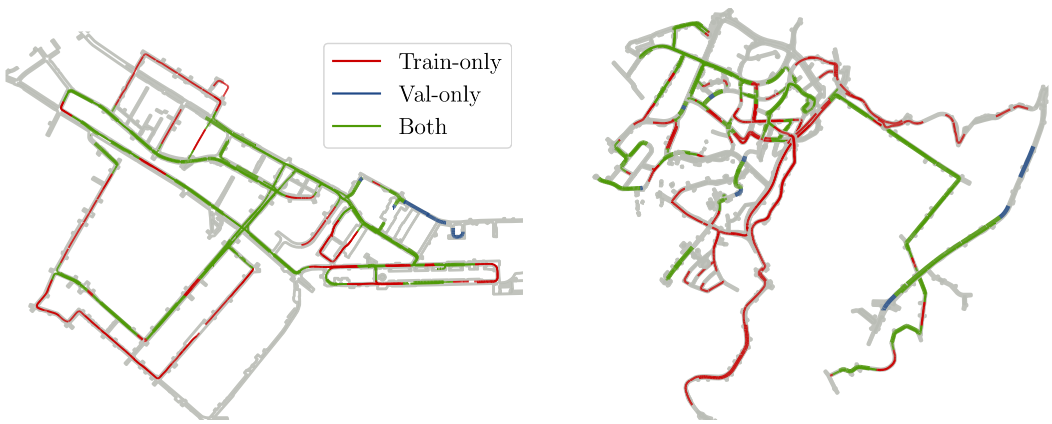
B.1 Hyperparameter Configuration

We provide the experimental configuration used in our main results. Table 7 lists the key hyperparameters for our compressed map prior implementation and training procedure.

Param	Value	Notes
T	2^{16}	Embedding size per level
L	4	Hash resolution levels
d	8	Embedding dimension
α_i	1.0-25.0	Hash level resolutions
η	2×10^{-4}	Base learning rate
$\beta_{1,2}$	$0.9, 0.999$	Adam betas
λ	0.01	Weight decay
N_{mask}	0.25	Prior masking ratio
B	8	Batch size
E	24	Training epochs
MLP	$[32, 32, 128]$	Projection dimensions
$\mathbf{X}_{\text{prior}}$	$\mathbb{R}^{h \times w \times 128}$	Prior dimensions
δ	50m	Proximity threshold

Table 7: Hyperparameter Configuration

B.2 Traversal Analysis Methodology



(a) Boston

(b) Singapore

Figure 8: **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.

We define traversal frequency as the number of distinct training scenes that overlap with a specific validation sample location. To calculate this metric, we process the nuScenes dataset through the following steps:

First, we extract all scene samples with their corresponding timestamps t_i, position vectors $p_i \in \mathbb{R}^3$, and transformation matrices to the global coordinate frame from the nuScenes dataset. Due to data collection procedures, continuous trajectories are sometimes fragmented across multiple scene recordings. We merge related trajectory fragments using temporal proximity ($\Delta t < 10s$) and spatial proximity ($\|p_{i,end} - p_{j,start}\| < 10m$) thresholds to create contiguous scenes that better represent continuous traversals.

After the trajectories are merged, for each sample we count the number of unique training scenes containing at least one sample within a 50-meter radius. See fig. 9 for the full distribution.

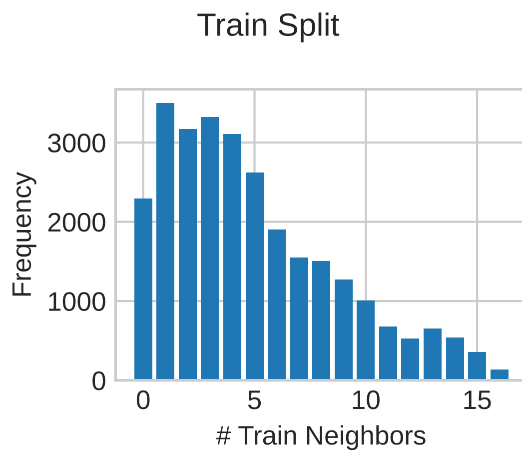


Figure 9: Distribution of traversal counts across the train split.

C USE OF LARGE LANGUAGE MODELS

In this work, LLMs were used for minor rephrasing and spell-checking assistance, specifically in the ablation analysis. We do not trust LLMs enough to use them in the results section.