Supplementary Material for *Spatial Retrieval Augmented Autonomous Driving*

Xiaosong Jia[1,2*], Chenhe Zhang[1,2*], Yule Jiang[3*], Songbur Wong[3*],
Zhiyuan Zhang[3], Chen Chen[4], Shaofeng Zhang[5], Xuanhe Zhou[3], Xue Yang[3†],
Junchi Yan[3†], Yu-Gang Jiang[1,2]

1. Institute of Trustworthy Embodied AI, Fudan University
2. Shanghai Key Laboratory of Multimodal Embodied AI
3. Shanghai Jiao Tong University
4. Key Laboratory of Target Cognition and Application Technology,
Aerospace Information Research Institute, Chinese Academy of Sciences
5. University of Science and Technology of China

[*]Equal contributions [†]Correspondence authors

Supplementary Material

Contents

A. Discussion

A.1. How does the latency of spatial retrieval influence the system?

In our experimental pipeline, all geographic images are retrieved, aligned, and processed *offline* before training and evaluation. This ensures that, during inference, the stack requires no interaction with external services (i.e. Google Maps APIs).

For real-world autonomous driving deployments, the geographic data can similarly be pre-downloaded and cached on the vehicle or on edge servers, as the storage cost is low due to the equirectangular panoramic representation that stores each street view location only once. Considering the required storage footprint is modest, the data can be bulk-downloaded during periods of reliable connectivity, allowing the inference stage to operate without relying on real-time network access and without being affected by bandwidth fluctuations.

Importantly, as demonstrated in our experiments, in cases where geographic images are unavailable during driving—such as regions with insufficient coverage or operation in offline-only settings—the Reliability Estimation Gate automatically down-weights or suppresses the geographic branch. The system thus reverts to the baseline perception pipeline without compromising safety or prediction quality, maintaining a stable and predictable lower-bound performance. Overall, the combination of offline acquisition, compact storage design, and modest computation makes the paradigm suitable for realistic deployment scenarios while avoiding latency overhead.

A.2. How does the method behave when the spatial retrieval data are missing, outdated, or unreliable?

Geographic images are inherently heterogeneous in density and recency, and discrepancies may arise due to construction, occlusions at capture time, or environmental changes. The proposed framework explicitly accounts for such imperfect conditions through the Reliability Estimation Gate, which evaluates both spatial distance and visual similarity between onboard and retrieved features. When retrieved images contain mismatched content, outdated structures, or notable photometric differences, the gate attenuates their contribution, preventing unreliable geographic cues from degrading the prediction.

The degradation experiment (Fig. 11 in the main paper) validates the robustness of the system by simulating missing or incorrect retrieval data. As the proportion of unavailable or perturbed data increases, performance declines smoothly and without catastrophic failure, demonstrating that those tasks remain supported by onboard-sensor alone. In other words, the model does not become overly dependent on geographic information and incorporates it only when reliable. This balanced fusion behavior ensures that the paradigm remains stable across a wide range of real-world conditions, including sparse coverage, partial misalignment, or outdated images.

A.3. Why do some tasks benefit more moderately from geographic context?

The geographic modality predominantly captures static environmental structure—road geometry, lane topology, sidewalks, and background scene layout—while lacking dynamic elements such as moving vehicles or pedestrians. Tasks that rely on static spatial organization, such as online mapping or occupancy prediction, naturally gain more from this prior. Improvements arise from the ability of geographic images to provide visibility into regions that may be occluded, poorly lit, or beyond the instantaneous perception horizon of onboard sensors.

For tasks whose performance is dominated by dynamic-object interpretation—such as object detection—the contribution of geographic priors is inherently more limited. The spatial retrieval paradigm is neither intended nor designed to replace online sensor information in such settings, but rather to serve as a reliable auxiliary cue. The more moderate gains observed for these tasks align with the role of static geographic information as an additive structural prior rather than a dynamic-sensing modality. This task-dependent pattern is consistent with the modality's characteristics and supports the intended complementary nature of spatial retrieval.

A.4. Is the proposed paradigm restricted to Google Maps as the source of geographic images?

Although Google Maps provides a convenient and accessible source of geographically anchored images for our experiments, the proposed paradigm is agnostic to the data provider. The method requires only panoramic or perspective images accompanied by approximate capture coordinates. Thus, it can be directly applied to images acquired through alternative mapping platforms or to large-scale traversal logs collected by autonomous driving fleets.

In practical deployments, companies may employ their own high-resolution street-level datasets, which often feature denser coverage and more consistent capture conditions than public sources. Because the retrieval function, panoramic representation, and fusion module rely only on pose and appearance alignment, no architectural modifications are required to accommodate different geographic data sources. This highlights that the core contribution lies in the spatial retrieval paradigm itself, rather than characteristics of any particular geographic images provider.

B. More Related Work

B.1. Camera-based 3D Detection

Bird's-eye-view (BEV) object detection [48, 87, 107, 111, 113, 145, 155, 166, 168] aims to detect objects in BEV space from single-view or multi-view 2D images. Early works [118, 120, 144] transform 2D features into BEV based on single-view images for monocular 3D detection. LSS [113] extends this to multi-view input and lifts image features into 3D via depth estimation. Following LSS, BEVDet [48] lifts 2D features to BEV and applies a BEV encoder with residual blocks and FPN, while BEVDepth [75] shows that explicit depth supervision and efficient BEV Pooling [47, 48, 75] are crucial for both accuracy and efficiency in the 2D-to-3D lifting pipeline. In contrast, BEVFormer [168] introduces a spatio-temporal Transformer encoder that directly constructs BEV representations from multi-view, multi-timestamp inputs via deformable cross-attention, and BEVFormerV2 [155] further eases optimization with perspective-view supervision and joint monocular/BEV supervision. From the view transformation perspective, BEVFormer and its variants [149, 155, 168] adopt dense BEV queries with heavy deformable attention, whereas BEVDet/BEVDepth [48, 75] follow an LSS-style [113] 2D-to-3D lifting plus BEV pooling pipeline. Inspired by DETR, sparse query-based BEV methods [87, 88, 145] instead start from a small set of 3D queries and interact with image features either via 3D-to-2D projection [145] or global attention with 3D positional embeddings [87, 88], trading dense BEV grids for a compact query set but often at the cost of expensive global attention and limited scalability to long-term temporal fusion.

B.2. Online Mapping

Online mapping models [69, 83, 89, 117, 161] tackle autonomous driving by dynamically generating map geometry and semantics from sensor inputs, thereby reducing reliance on manual annotations and enabling adaptive mapping in changing environments. With the development of PV-to-BEV methods [66, 96], HD map construction is often formulated as BEV segmentation on surround-view images, where many approaches [15, 40, 90, 91, 108, 113, 116, 167–169] produce rasterized maps through BEV semantic segmentation. To obtain vectorized HD maps, HDMapNet [69] performs heuristic and time-consuming post-processing on pixel-wise segmentation, while VectorMapNet [89] is the first end-to-end method with a two-stage coarse-to-fine design and an autoregressive decoder, suffering from long inference time and permutation ambiguity. MapTR [83] and StreamMapNet [161] further leverage monocular or multi-view images for end-to-end online HD map construction, enhancing deployment flexibility, but most existing models are trained on fixed datasets and tightly coupled to specific sensor setups and environments, leading to overfitting and degraded performance under domain shift. SemVecNet [117] mitigates cross-dataset variation via an intermediate semantic map, yet still falls short of the robustness required for real-world deployment. In contrast, our method introduces a unified shape modeling strategy for map elements that resolves permutation ambiguity and stabilizes training, and further builds a structured, parallel one-stage framework with much higher efficiency. Concurrent and follow-up works of our conference version [83] explore alternative end-to-end HD map constructions [11, 12, 57, 58, 71, 72, 92, 114, 115, 123, 141, 151, 153, 161–163] and extend to related tasks [16, 54, 55, 70, 82, 152], including Bézier- and pivot-based map representations [24, 114], neural global map representations [152], topology-aware modeling [70], diffusion-based refinement [11], differentiable rasterization [163], and generalized centerline and 3D map learning in MapTRv2.

B.3. Occupancy Prediction

Occupancy prediction has recently emerged as a 3D alternative to conventional bird's-eye view (BEV) perception. BEV representations [67, 97] provide an occlusion-reduced and metrically consistent top-down view that is well suited for multi-view, multimodal, and temporal fusion, as well as downstream planning and decision making. However, BEV inevitably collapses the height dimension and thus cannot fully capture fine-grained 3D geometry or vertical structures. Occupancy perception addresses this limitation by estimating the occupied state of voxels in a discretized 3D space. Such dense 3D fields naturally support open-set objects, irregular shapes, complex road structures, and occlusion reasoning [1, 146], and can be jointly used for 3D detection, segmentation, and tracking [103, 112, 146, 170]. Recent work further enriches occupancy with semantics [131], language priors [136], and motion cues [95], making vision-centric occupancy a cost-effective alternative to LiDAR-heavy systems.

In the camera-only setting, most methods still build on BEV-style formulations to infer voxel-wise occupancy from multi-view images. One line of work directly learns voxel features in 3D space: VoxFormer [76] uses 2.5D cues to initialize voxel queries, Occ3D [130] adopts a coarse-to-fine voxel encoder, and RenderOcc [109] predicts density and semantics under NeRF-style supervision, facilitated by dense occupancy benchmarks [124, 130]. Another line couples occupancy prediction more tightly with BEV representations. TPVFormer [49] decomposes the scene into three orthogonal views, Sur-

roundOcc [148] lifts BEV features along the height axis with spatial cross-attention, OccNet [132] builds a unified occupancy embedding tailored for planning, and FBOcc [80] designs a forward–backward view transformation scheme.

B.4. End-to-End Planning

Autonomous driving planning has evolved through several distinct stages, beginning with rule-based frameworks and gradually shifted toward learning-driven and end-to-end systems. Early planning modules were dominated by rule-based methods, where vehicle behavior followed manually designed rules and heuristic decision logic [23, 28, 133]. These planners offered strong interpretability and reliable control, and were successfully deployed in many real-world systems [61, 134]. Their fundamental limitation, however, lies in their inability to generalize: once the environment deviates from predefined scenarios, the decision-making process becomes brittle.

To improve adaptability, the community gradually shifted toward learning-based planners, framing planning as an imitation learning problem [13, 44, 129]. Early behavior cloning approaches relied on CNNs [5, 39, 59] and RNNs [4], and were later extended to Transformer-based architectures [20, 122] to better capture complex driving behaviors. While these models produce more human-like trajectories, imitation learning lacks multi-modality guarantees and is sensitive to distribution shift, resulting in compounding errors during closed-loop execution. Consequently, many systems still fall back on rule-based post-processing—either refining [51, 135] or selecting [19] trajectories—which runs counter to the original goal of eliminating handcrafted rules. Moreover, tuning behavior for safety or personalization is difficult, since training-time objectives often conflict with one another [4, 19].

Driven by the need for tighter integration between perception, prediction, and planning, planning has increasingly been studied under the umbrella of end-to-end autonomous driving (E2EAD). Modern end-to-end planners directly map sensor inputs to future trajectories or control actions [17, 18, 42, 44, 53, 55, 56, 77, 79, 121, 125, 126, 128], reducing the error accumulation inherent in multi-stage pipelines. UniAD [44] established a unified planning-oriented architecture; VAD [55] and VADv2 [17] streamlined the representation and action space; SparseDrive [128] leveraged sparse structures; and diffusion-based models such as DiffusionDrive [84] introduced generative planning via diffusion policies.

To further improve planning safety, recent efforts have explored score-based planning optimization. Hydra-MDP [81] distills both human demonstrations and rule-derived metrics through multiple scoring heads, while WOTE [78] predicts future BEV states to evaluate trajectory quality. These approaches enhance safety supervision but still operate at the score level, independently optimizing each anchor without modeling the global policy distribution.

B.5. Generative World Model

Recent progress in driving video generation has advanced realistic and controllable autonomous driving simulation [31, 32, 34, 62, 63, 100, 139, 147]. MagicDrive [32] achieves high-fidelity street-view synthesis through tailored encoders and cross-view attention for accurate 3D geometry control, while UniScene [64] unifies multi-modal data generation (semantic occupancy, video, LiDAR) via progressive processes. Extensions such as MagicDriveDiT [31] and MagicDrive3D [30] further improve and enhance scalability using DiT architectures and deformable Gaussian splatting for high-resolution 3D reconstruction. DreamDrive [100] synthesizes 4D spatiotemporal scenes via hybrid Gaussian representations, balancing visual fidelity and generalizability.

Beyond open-loop video synthesis, a related line of work explores using generative models as driving world models that predict future multi-view observations conditioned on controls and scene layouts. Unlike general-purpose video generation, generative world models for autonomous driving are designed to support multi-agent interaction, ego-motion control, environmental diversity, and coherent multi-camera observations. Early efforts in this direction include GAIA-1 [41], which introduces text and ego-action conditioning within a discrete world model equipped with a video diffusion decoder to enhance temporal consistency, and CommaVQ [21], which similarly employs a causal transformer over discrete tokens to model controllable ego-motion in driving scenes.Subsequent approaches build on latent diffusion models to achieve higher-fidelity generation and richer forms of control. DriveDreamer [143] conditions a latent diffusion model on 3D bounding boxes, HD maps, and ego actions, and further adds an action decoder for future ego-action prediction. Drive-WM [147] extends this paradigm to multi-camera settings with up to six surrounding views and introduces controllable ego behavior, dynamic agents, and environmental factors such as weather and lighting. UniMLVG [14] enables multi-view video synthesis conditioned on text, camera parameters, 3D bounding boxes, and HD maps, while MaskGWM [106] towards longer temporal horizons. Vista [33] focuses on high-resolution, long-duration videos, and Delphi [94] steers generation toward failure-prone scenarios to enhance the utility of synthetic data for training.

B.6. Autonomous Driving with Retrieval

Recent research in autonomous driving has increasingly emphasized retrieval-augmented decision making, where external knowledge is queried and combined with on-board perception rather than relying solely on end-to-end models. Neuro-symbolic approaches [157] and knowledge-graph-based methods [27, 101] can be seen as early forms of retrieval, using symbolic structures to recall missing entities or relations (e.g., occluded pedestrians or lane-change intentions). In parallel, cooperative perception retrieves observations from other agents in the traffic network via V2X communication [37, 86, 119, 156], mitigating the "short-sightedness" of purely on-board perception [102] and improving situation awareness, especially in occluded or dense scenes [93, 104, 150]. Recent work such as CodeFilling [45, 127, 137] further moves from passive broadcast to more selective, representation-aware communication, which conceptually aligns with retrieval-style filtering.

More directly, a growing body of work treats driving scenarios as the core memory to be retrieved in Retrieval-Augmented Generation (RAG) pipelines. Structured scenario platforms (OpenScenario, MetaScenario, CommonRoad) [2, 3, 8] and large-scale datasets [8, 10, 25, 29, 35, 43, 46, 50, 65, 73, 74, 165] provide the basis for retrieving past interactions and corner cases. LLM-based Driving-RAG systems query similar scenarios and textual knowledge to support online planning and decision making [22, 52, 154, 160] and offline scenario generation and simulation [36, 105, 164], while other works use LLMs with retrieved driving knowledge for instruction following and human–vehicle interaction [7, 9, 140]. To make such retrieval effective, prior studies propose task-driven scenario embeddings [98, 138, 142] and similarity metrics [10, 38, 60, 171], sometimes combined with distribution-aware indexing [43, 68, 99] and graph-based post-processing [110, 138] to refine the top-K retrieved scenarios and reduce LLM confusion and hallucination.

C. nuScenes-Geography Construction Details

C.1. Overview of the Construction Pipeline

The objective of nuScenes-Geography extension is to augment each nuScenes frame with geographically grounded visual information retrieved from the Google Maps APIs, which include both the Google Street View API and the map-tile interfaces. Among these sources, street view images constitute the core component of our pipeline due to its need for viewpoint synthesis and geometric alignment; thus, it is the primary focus of the following sections. In contrast, map data follow a significantly simpler procedure: we download the complete nuScenes regional map beforehand and crop the corresponding regions directly based on the transformed coordinates, requiring no additional viewpoint modeling.

The overall construction pipeline contains three major stages. First, we convert nuScenes ego poses from their local map coordinate system into global geodetic coordinates (latitude–longitude), which enables precise querying of the Google Maps APIs. Second, for each valid geographic location, we retrieve multi-view street view images via the Google Street View API and reconstruct an equirectangular panoramic representation that compactly encodes all viewing directions. Finally, for every nuScenes camera frame, we instantiate a virtual camera at the matched street view location and synthesize a geometrically aligned street-level view through spherical-to-perspective reprojection. This pipeline establishes a one-to-one correspondence between each onboard camera image and its aligned street view rendering, while maintaining high storage efficiency and broad geographic coverage across the dataset.

C.2. Coordinate Systems and Geo-Localization

Ego-pose representation in nuScenes. We adopt the nuScenes global coordinate frame as the starting point for geo-localization. For each driving scene, we follow the temporal sequence defined by the `LIDAR_TOP` sensor and obtain a set of ego poses $\{\mathcal{P}_i\}_{i=1}^{N}$. Each pose $\mathcal{P}_i$ consists of a 3D translation $\mathbf{t}_i = (x_i, y_i, z_i)^\top$, a unit-quaternion rotation $\mathbf{q}_i$, and a timestamp t_i. Within each nuScenes location (e.g., `boston-seaport`), the global coordinates provide a consistent, locally planar map in which horizontal translations (x_i, y_i) describe the vehicle's trajectory.

Geodetic reference and modeling assumptions. To relate this local map to Earth-centered geographic coordinates, we introduce a reference latitude–longitude pair $(\phi^{\ell}_{\text{ref}}, \lambda^{\ell}_{\text{ref}})$ for each location ℓ. The reference point is chosen near the south-western boundary of the nuScenes map and serves as the anchor of a local tangent plane. Such planar georegistration is a common convention for urban-scale coordinate systems, and because each nuScenes region spans only a few thousand meters, the induced geodesic distortion is negligible.

From planar displacements to geodetic coordinates. Given an ego pose $\mathcal{P}_i$, its horizontal translation (x_i, y_i) is treated as a displacement on the local tangent plane. We compute its ground-plane distance

$$d_i = \sqrt{x_i^2 + y_i^2}, \tag{1}$$

and its azimuth using the two-argument arctangent

$$\theta_i = \mathrm{atan2}(x_i, y_i), \tag{2}$$

which correctly resolves the angular quadrant. Starting from the reference coordinate $(\phi^{\ell}_{\mathrm{ref}}, \lambda^{\ell}_{\mathrm{ref}})$, we apply a forward geodesic model

$$(\phi_i, \lambda_i) = F\big(\phi^{\ell}_{\mathrm{ref}}, \lambda^{\ell}_{\mathrm{ref}}, \theta_i, d_i\big), \tag{3}$$

where $F(\cdot)$ performs the standard "move by distance d_i along initial bearing θ_i" operation on a WGS-84 ellipsoid. Within the spatial extent of nuScenes, the planar distance d_i provides an accurate approximation to the geodesic arc length.

Per-frame localization for geographic retrieval. Applying this transformation independently to every ego pose yields a set of georeferenced records $(t_i, \phi_i, \lambda_i, \mathbf{q}_i)$. The latitude–longitude coordinates (ϕ_i, λ_i) serve as query points for the Google Maps APIs, enabling frame-level association between nuScenes and geographic data.

C.3. Geographic Data Retrieval from Google Maps APIs

C.3.1. Street View Image Acquisition

Street View Metadata API vs. Image API. Our street view acquisition pipeline is built upon two complementary Google interfaces: the Street View *Metadata API* and the Street View *Image API*. Both APIs accept a geographic query location (ϕ_i, λ_i), but they differ fundamentally in purpose and returned information.

The Metadata API returns a structured description of the street view panorama geographically closest to the query point. Its response includes: (i) a success status flag, (ii) a *unique panorama identifier* that labels the corresponding street view panorama, (iii) the panorama's canonical capture position in latitude–longitude form, and (iv) auxiliary attributes such as capture date. If the query location is not covered by street view data, the response explicitly indicates that no panorama is available.

By contrast, the Image API returns a single perspective street view image rendered at a specified heading, pitch, and field of view. Its inputs include the geographic position, the viewing configuration, and the desired image resolution. The returned image corresponds to a projection of the underlying panorama identified by the Metadata API.

Because these two APIs provide different levels of information, we query the Metadata API first to determine whether a panorama exists and to obtain its unique identifier and canonical capture location. Only after this association is established do we use the Image API to request the multi-view perspective images required for panoramic reconstruction. This separation allows us to avoid unnecessary downloads and ensures that all perspective images are consistently tied to a single, well-defined street view panorama.

Metadata-driven panorama association and nearest-neighbor behavior. The Metadata API implicitly performs a nearest-neighbor search over Google's street view graph: each nuScenes coordinate (ϕ_i, λ_i) is mapped to the geographically closest existing street view panorama, if any. This produces a deterministic association

$$(\phi_i, \lambda_i) \;\longrightarrow\; \text{panorama index } \mathcal{P}_i.$$

However, because the street view sampling is sparse and road networks contain parallel lanes, service roads, and split-level structures, the nearest panorama may occasionally lie on a different road segment despite being close in Euclidean distance. These mismatches constitute a well-defined misaligned mode and motivate the reliability analysis presented later in Sec. D.3.

Unique panorama retrieval and caching. Once a panorama index $\mathcal{P}$ is identified by the Metadata API, we ensure that its underlying panoramic content is downloaded only once. When a panorama is encountered for the first time, we issue a set of Image API requests at the panorama's canonical capture location to obtain the multi-view images required for reconstructing its equirectangular representation. The reconstructed panorama is stored on disk and indexed by its identifier. Any subsequent frame whose Metadata query maps to the same $\mathcal{P}$ directly reuses the cached equirectangular panorama, eliminating redundant API calls. This mechanism exploits the natural many-to-one relationship between nuScenes frames and street view panoramas while maintaining computational and storage efficiency.

Figure 1. **Examples of Street View API outputs.** Eighteen sample images returned by the Street View APIs, illustrating panorama lookup and the multi-view perspective captures used for equirectangular reconstruction.

Multi-view sampling for equirectangular reconstruction. To reconstruct the full 360° panorama associated with $\mathcal{P}$, we sample a fixed set of uniformly spaced heading angles,

$$\Theta = \{0^\circ, 20^\circ, 40^\circ, \ldots, 340^\circ\},$$

and request a perspective street view image for each heading at the panorama's capture position. As illustrated in Fig. 1, every request uses a fixed horizontal field of view (60 degrees in our implementation), a fixed pitch configuration, and a fixed output resolution. Collectively, these multi-view images provide dense angular coverage around the street view camera and serve as input for the equirectangular projection and stitching process described in Sec. C.4.

Frame-to-panorama mapping structure. The combination of the Metadata API and Image API yields a compact, surjective mapping

$$\text{nuScenes frame } i \;\longrightarrow\; \text{panorama } \mathcal{P}_i,$$

where many nuScenes frames may share the same panorama. This mapping guarantees that each nuScenes frame is associated with a unique equirectangular street view panorama, and that all perspective renderings used in later stages (e.g., virtual camera synthesis) are derived from a single, geometrically consistent spherical representation.

C.3.2. Satellite images and Map Data

Overview. In addition to the street view panoramas, we incorporate high-resolution satellite images aligned with the nuScenes maps. Satellite images provides a static top-down representation of the environment and does not require the spherical modeling or equirectangular reconstruction used for street view.

Region of interest and spatial resolution. For each nuScenes location ℓ, we determine the spatial extent of its map in the local coordinate frame and anchor this extent to the geodetic reference point $(\phi^\ell_{\text{ref}}, \lambda^\ell_{\text{ref}})$ introduced in Sec. C.2. From this anchor, a geographic bounding region is defined that fully covers all trajectories within the corresponding nuScenes split. Satellite images for this region is requested at a ground sampling distance of

$$0.15 \text{ meters per pixel},$$

ensuring that each pixel in the resulting raster corresponds to approximately $0.15\,\text{m}$ on the ground.

Single-raster satellite mosaic per location. All satellite tiles covering the defined geographic region are aggregated into a single georeferenced mosaic for each location ℓ. This mosaic fully spans the nuScenes map footprint and serves as the unified satellite representation for all frames belonging to that location.

Per-frame, pose-aware satellite crops. For a nuScenes frame with geodetic coordinates (ϕ_i, λ_i) and an ego-vehicle orientation, we compute a pose-aligned satellite crop in two steps. First, the frame's geodetic position is transformed into pixel coordinates on the mosaic using the known geographic reference. This is a direct affine mapping from latitude–longitude to image coordinates. Second, a fixed-size satellite patch is extracted around the mapped pixel and rotated according to the ego-vehicle yaw angle. The rotation aligns the vehicle's forward direction with the *right-hand side* of the cropped satellite patch. Thus, each crop is both spatially centered at the vehicle location and oriented in accordance with the vehicle's heading, yielding a canonical, pose-aware top-down representation.

C.4. Equirectangular Panorama Construction

Motivation. Each street view panorama is conceptually a spherical environment map centered at the street view camera position. However, the Google Street View Image API only provides perspective renderings of this sphere at user-specified headings and pitch angles. To obtain a complete and consistent representation of the underlying spherical scene, we reconstruct an equirectangular panorama from multiple perspective views sampled around the panorama center.

Multi-view sampling of the street view sphere. Given the canonical capture location returned by the Metadata API, we sample the street view sphere at a fixed set of uniformly spaced heading angles,

$$\Theta = \{0^\circ, 20^\circ, \ldots, 340^\circ\},$$

and at a fixed pitch configuration. Each sample corresponds to a perspective projection with a fixed horizontal field of view and image resolution. These images form a collection of overlapping observations covering the full 360° horizontal field around the street view camera.

Spherical projection model. To integrate these perspective views into a single panorama, each image is interpreted as a collection of rays originating from the panorama center. For a pixel at coordinates (u, v) in a perspective image with known intrinsics, the corresponding 3D ray direction $\mathbf{d}$ is computed through inverse camera projection. This ray is then parameterized on the unit sphere using longitude–latitude angles

$$(\theta, \phi) \in [-\pi, \pi) \times [-\frac{\pi}{2}, \frac{\pi}{2}],$$

where θ is the azimuth and ϕ is the elevation. The equirectangular panorama is defined on a regular grid over (θ, ϕ), and each incoming ray contributes its color to the corresponding grid cell.

Equirectangular projection and rasterization. The spherical surface is discretized into a raster. For each heading–pitch configuration in the sampled set, the corresponding perspective image is projected onto this raster. This projection uses a forward-mapping formulation: the spherical direction (θ, ϕ) is mapped to pixel coordinates

$$x = \frac{W}{2\pi}(\theta + \pi), \qquad y = \frac{H}{\pi}\left(\frac{\pi}{2} - \phi\right),$$

and the color is written into the equirectangular grid. A binary mask records which pixels receive valid contributions, ensuring that overlap regions from multiple views can be correctly combined.

Mask-based panoramic stitching. The equirectangular panorama is assembled by compositing all projected views. For each view, only pixels whose rays fall within the valid forward-facing region are retained; invalid regions (e.g., back-facing directions or watermark areas) are discarded using the mask.

Resulting spherical representation. This produces a dense, seamless panorama encoding the full horizontal surround view, as illustrated in Fig. 2.

This panorama serves as the unique, spherical representation associated with each street view location and constitutes the basis for synthesizing nuScenes-aligned virtual camera views in Sec. C.5.

Figure 2. **Examples of reconstructed equirectangular panoramas.** Five representative panoramas generated using our multi-view sampling procedure. Each panorama is produced by aggregating multiple overlapping street view perspective images sampled at uniformly spaced heading angles. The resulting equirectangular representations are dense and seamless, serving as the canonical environment map for subsequent virtual camera synthesis.

C.5. Virtual Camera Model and Panorama Reprojection

Overview. To obtain street view images that matches the viewing geometry of nuScenes camera frames, we construct for each frame a virtual camera located at the street view panorama position. Its orientation follows the calibrated nuScenes camera, while its intrinsic parameters follow a unified perspective model consistent with the rendering geometry used during panorama extraction. A view aligned with this virtual camera is then synthesized by sampling the panorama.

Intrinsic model. All virtual cameras share a fixed horizontal field of view (approximately 60°) and a square output resolution (W, H). The focal length is defined as

$$f_x = \frac{W}{2\tan(\mathrm{FOV}/2)}, \qquad f_y = f_x, \tag{4}$$

with the principal point placed at the image center,

$$c_x = \frac{W}{2}, \qquad c_y = \frac{H}{2}. \tag{5}$$

Figure 3. **Panorama projection and corresponding onboard views.** The top row shows the equirectangular panorama associated with a street view location. The middle row presents six perspective views obtained by projecting the panorama through the virtual camera model in Sec. C.5. The bottom row displays the corresponding six nuScenes onboard camera images. These examples illustrate that the projection and reprojection steps provide a well-aligned geometric correspondence across the panorama domain and the nuScenes camera domain.

The resulting intrinsic matrix is

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

This formulation matches the projection geometry used for the panorama construction and ensures consistent spherical–perspective mapping.

Extrinsic construction. The virtual camera is positioned at the street view capture location associated with each nuScenes frame. The displacement between the nuScenes ego location and the street view panorama is computed in geodetic coordinates and converted into a metric offset on a local plane. The vertical component is set to a constant height of $2\,\mathrm{m}$. To express this translation in the ego-vehicle coordinate system, the offset is rotated by the inverse of the ego orientation. The rotation of the virtual camera is taken from the calibrated nuScenes camera for the corresponding sensor. Together, these parameters define a 4×4 camera pose used for spherical reprojection.

Inverse projection to rays. For each output pixel (u, v), the corresponding camera ray is computed using the intrinsic matrix. Let

$$X = \frac{u - c_x}{f_x}, \qquad Y = \frac{v - c_y}{f_y}, \qquad Z = 1, \tag{7}$$

and normalize to obtain a unit direction:

$$\mathbf{d}(u, v) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{8}$$

Applying the virtual camera rotation yields the ray direction in the panorama coordinate frame.

Spherical mapping and sampling. Given the rotated ray $\mathbf{d}' = (X', Y', Z')^\top$, we compute its spherical azimuth and elevation as

$$\theta = \arctan 2(X', Z'), \qquad \phi = \arcsin(Y'). \tag{9}$$

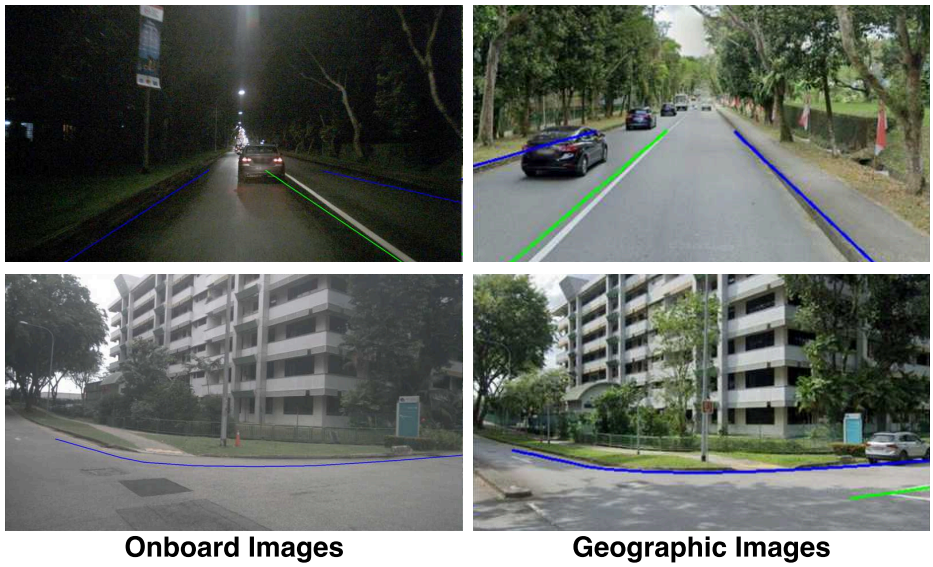

Figure 4. **Geometric consistency of lane reprojection.** Visualization of structural annotations reprojected into the aligned street view image using the virtual camera model in Sec. C.5. The observed spatial alignment between the overlaid geometric curves and the roadway appearance suggests that the overall mapping and reprojection pipeline produces a well-aligned geometric correspondence across domains.

For an equirectangular panorama of width W_{pano} and height H_{pano}, the corresponding pixel coordinates are

$$x = \frac{\theta + \pi}{2\pi} W_{\text{pano}}, \qquad y = \frac{\frac{\pi}{2} - \phi}{\pi} H_{\text{pano}}. \tag{10}$$

Sampling the panorama at (x, y) produces the color of pixel (u, v) in the synthesized street view image.

Aligned street view viewpoints. The resulting image represents the street view environment as observed from the nuScenes camera orientation, as shown in Fig. 3. By applying this process to each nuScenes frame, we obtain a geometrically aligned street view counterpart for every camera observation, providing a consistent basis for spatial correspondence and cross-modal retrieval. As shown in Fig. 4, this projection mechanism produces a geometrically accurate alignment between the reprojected structures and the street view imagery.

C.6. Retrieval Misalignment and Manual Reliability Annotation

Overview. Although the Google Street View Metadata API performs a nearest-neighbor search in geographic space and typically returns the panorama closest to a queried nuScenes location, the retrieved panorama is not guaranteed to correspond to the viewpoint captured by the nuScenes camera. Due to the sparsity and heterogeneous spatial distribution of street view coverage, mismatches arise in several systematic forms. To ensure the integrity of our aligned dataset and to enable supervised learning of retrieval reliability, we manually annotate all API-returned panoramas associated with nuScenes frames.

Misaligned modes of API-returned panoramas. In practice, incorrect panorama associations arise from the following well-defined misaligned modes:

- **Indoor or non-road panoramas.** The API may return panoramas captured in buildings, parking structures, or enclosed environments, even when the nuScenes frame is recorded on a public road.
- **Incorrect vertical level (bridge–ground confusion).** For multi-level infrastructure, the nearest panorama may lie at the wrong elevation, showing the underside of a bridge when the vehicle is driving above it, or the reverse.
- **Parallel-road ambiguity.** Dense urban areas often contain parallel roads only a few meters apart. Proximity-based retrieval may incorrectly return a panorama from the adjacent roadway, leading to consistent semantic and structural mismatch.

As shown in Fig. 5, these misalignment patterns constitute the primary sources of retrieval noise in the nuScenes–street-view pairing, motivating the need for a systematic annotation procedure.

Figure 5. **Representative misaligned cases of API-returned panoramas.** Each row shows a mismatch example, where the *top* image is the corresponding nuScenes onboard camera view and the *bottom* image is the retrieved street view panorama. From top to bottom, the three cases illustrate: (1) a non-road panorama returned for an outdoor driving frame, (2) an incorrect vertical level due to bridge–ground confusion, (3) a parallel-road mismatch where the panorama lies on an adjacent roadway.

Manual reliability annotation. Each retrieved panorama is manually compared with the corresponding nuScenes frame. Annotators inspect the panorama in its reconstructed equirectangular form and the nuScenes image in its native camera geometry. The annotation task is strictly comparative and relies only on observable scene-level cues, including roadway configuration, building geometry, structural alignment, and global viewing direction.

To provide clean and unambiguous supervision, we adopt a ternary labeling scheme:

- **Reliable (positive).** Assigned when the panorama and the nuScenes image exhibit clear semantic and geometric consistency, with matching road layout, scene structure, and approximate orientation, as shown in Fig. 6.
- **Unreliable (negative).** Assigned when the panorama clearly corresponds to a different physical environment or viewpoint, such as incorrect elevation, adjacent roads, or indoor locations, as shown in Fig. 5.
- **Unlabeled (ambiguous).** Used for cases where the correspondence cannot be judged confidently—for example, when the panorama is geographically close but exhibits subtle structural differences not resolvable from the nuScenes view. These samples are excluded from all supervised training.

This design provides a controlled supervisory signal by retaining only cases with unambiguous alignment or misalignment, thereby ensuring learnability.

C.7. Temporal Distribution of Retrieved Street View Images

The Google Maps Street View API returns its most recently captured panorama for each queried location. As a result, the capture dates of retrieved street view images span a range of years, while the nuScenes dataset was collected in 2018. Fig. 7

Figure 6. **Examples of reliable positive matches.** The figure presents two positive cases, each consisting of a street view panorama (top) and its corresponding nuScenes onboard camera image (bottom). In both cases, the two modalities exhibit consistent scene structure, including roadway layout, surrounding buildings, and overall orientation. These clear and coherent correspondences are used as reliable positive supervision in our analysis.

shows the temporal distribution of the retrieved street view images. Despite the temporal mismatch between the street view capture dates and the nuScenes recording period, the proposed Reliability Estimation Gate (REG) effectively mitigates the impact of outdated imagery by down-weighting features from temporally misaligned retrievals.

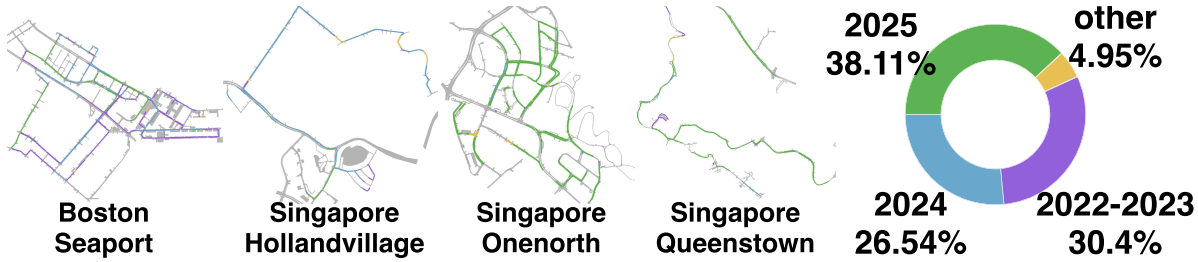

Figure 7. **Temporal distribution of retrieved street view images.** The capture dates of street view panoramas span from 2016 to 2025, while the nuScenes dataset was collected in 2018.

D. Implementation Details of Neural Networks

D.1. 3D Postional Encodings for Retrieved Geographic Images

To incorporate retrieved street-view and satellite images into BEV-based representations in a geometrically consistent manner, we adopt the positional encoding mechanism introduced in PETR [87]. Geographic images provide valuable spatial priors, but their feature maps reside in the image plane and do not contain explicit 3D information. PETR enables the construction of geometry-aware positional encodings that describe the spatial coordinates associated with each pixel location, thereby facilitating cross-view interaction with BEV features.

Depth Sampling and Coordinate Generation. For each geographic image, a discrete set of depth values $\{d_k\}_{k=1}^{D}$ is defined within the valid operating range of the dataset. Each pixel coordinate (u, v) is then associated with a three-dimensional

point computed via inverse projection:

$$\mathbf{x}_{uvk} = \pi^{-1}(u, v, d_k),$$

where π^{-1} applies the camera intrinsics and the corresponding street-view or satellite camera pose. This procedure generates a dense set of 3D coordinates $\mathbf{x}_{uvk}$ for all pixels and depth samples.

Normalization and Positional Encoding. Following PETR, the 3D coordinates are normalized to a fixed range $[x_{\min}, y_{\min}, z_{\min}] - [x_{\max}, y_{\max}, z_{\max}]$ to ensure numerical stability:

$$\tilde{\mathbf{x}}_{uvk} = \frac{\mathbf{x}_{uvk} - \mathbf{x}_{\min}}{\mathbf{x}_{\max} - \mathbf{x}_{\min}}.$$

The depth dimension is concatenated along the channel axis, after which a lightweight 1×1 convolutional mapping is applied:

$$\mathbf{P}_{\text{geo}} = f_\theta(\tilde{\mathbf{x}}),$$

producing a geometry-aware positional embedding aligned spatially with the retrieved geographic feature map.

Street View and Satellite Encoding. For retrieved street-view images, the full PETR-style depth frustum is used, based on the camera intrinsics and extrinsics stored in the metadata. For satellite images, which typically corresponds to a near-planar observation of the environment, pixel locations are mapped to the LiDAR coordinate system via a specific 2D transformation, with the vertical coordinate set to $z = 0$. Both cases employ the same positional encoder f_θ, allowing a unified geometric representation across modalities.

Integration into Cross-Attention. The positional encoding $\mathbf{P}_{\text{geo}}$ serves as the geometry-aware key in the Geo-Cross-Attention module:

$$\text{CrossAttn}(\mathbf{F}_{\text{BEV}}, \mathbf{F}_{\text{geo}} + \mathbf{P}_{\text{geo}}, \mathbf{F}_{\text{geo}}),$$

enabling BEV queries to attend to retrieved geographic features according to their spatial correspondence rather than solely on appearance. This positional encoding ensures that retrieved geographic features are integrated into BEV space in a manner consistent with the underlying 3D structure.

D.2. Spatial Retrieval Adapter

Our Spatial Retrieval Adapter is designed to be highly flexible and can be integrated into various BEV perception models.

BEVFormer-based Architecture For architectures that are inherently attention-based, such as BEVFormer [168], the integration is particularly natural. The BEVFormer encoder consists of stacked layers, each performing temporal self-attention and spatial cross-attention. We simply insert our **Geo-Cross-Attention** module at the end of each of these layers. This allows the model to progressively refine its BEV representation at each stage of the encoder, first using onboard sensor data and then enriching it with the retrieved global context, while keeping the rest of the architecture unchanged.

Other BEV Architectures. The adapter's plug-and-play nature allows for flexible application in other models, such as those based on Lift-Splat-Shoot (LSS) [113]. The core principle is to apply the Geo-Cross-Attention module at any stage where a BEV feature map is present. We demonstrate this flexibility with two examples:

- For **BEVDet** [48], which processes BEV features at multiple scales, we apply our adapter after each downsampling stage in the BEV backbone. This enables multi-scale fusion of the geographic context.
- For **Flash-Occ** [159], to accommodate its significant GPU memory requirements, we apply the Geo-Cross-Attention module only once, on the final and most compact BEV feature map (100×100 resolution). This highlights the adaptability of our method to different computational and memory constraints.

Generative Architecture. Existing generative world models [31, 158] for autonomous driving are mostly built by fine-tuning pretrained generative models [26]. Concretely, the original MM-DiT backbone is frozen, and trainable temporal as well as spatial cross-attention modules are added to the DiT blocks to enhance multi-view and multi-frame consistency. Our geo-adapter is inserted after the spatiotemporal cross-attention and uses zero-initialized convolutions together with gated linear layers to ensure minimal disturbance to the original model.

These examples underscore that our adapter is not tied to a specific architecture but can be strategically placed to balance performance and efficiency.

D.3. Reliability Estimation Gate

Geographic retrieval unavoidably suffers from missing, outdated, or misaligned street-view images. Such imperfections introduce uncertainty in the external priors and can negatively affect downstream perception. To ensure that the model incorporates geographic information only when it is trustworthy, we employ a *Reliability Estimation Gate*, which produces a confidence score $w \in [0, 1]$ that modulates the contribution of retrieved features during fusion. The gate combines both appearance-level and geometry-level cues, enabling the system to handle noisy geographic coverage.

Feature Similarity via ZNCC. The primary indicator of reliability is the visual consistency between current onboard observations and the retrieved street-view features. Given an onboard feature map $\mathbf{F}_{\text{on}}$ and a retrieved feature $\mathbf{F}_{\text{geo}}$, the latter is first aligned to the onboard spatial resolution through scale-preserving interpolation followed by center cropping. A Zero-Normalized Cross-Correlation (ZNCC) operator is then applied locally to measure their photometric agreement:

$$\text{ZNCC}(x, y) = \frac{\mathbb{E}[(x - \mu_x)(y - \mu_y)]}{\sqrt{\mathbb{E}[(x - \mu_x)^2]\,\mathbb{E}[(y - \mu_y)^2]} + \epsilon},$$

where x and y denote corresponding patches with 9×9 kernel size and μ_x, μ_y their local means. We convert ZNCC ranges in $[-1, 1]$ into a normalized difference measure,

$$\text{Diff} = \frac{1 - \text{ZNCC}}{2} \in [0, 1],$$

which provides a spatially dense signal reflecting how much the retrieved appearance deviates from the real-time observation. Larger values indicate less trustworthy geographic information.

Distance-Based Consistency. Street view images far from the ego location are more likely to reflect a different environment or capture outdated structures. To account for this, we encode the GPS distance d_{GPS} between the retrieved viewpoint and the current position. Instead of using the raw distance—which can become numerically unstable—we apply a bounded mapping:

$$d' = \tanh\left(\frac{d_{\text{GPS}}}{s}\right),$$

where s is a fixed scaling factor. This produces a distance cue $d' \in [-1, 1]$ that effectively suppresses geographically irrelevant retrievals without over-penalizing small, benign spatial offsets.

Reliability Prediction. The visual difference map and distance cue are aggregated to compute the final reliability score:

$$w = \sigma\left(f_\theta(\text{Diff}, d')\right),$$

where f_θ denotes a trainable prediction function and σ is the sigmoid activation. The resulting w serves as an adaptive weight during fusion, increasing the impact of geographically consistent retrievals and down-weighting those that are mismatched or outdated.

Training Supervision. During training, the gate is supervised with binary labels (detailed in C.6) indicating whether a retrieved sample is valid (1) or invalid (0). The loss is implemented using a binary cross-entropy objective:

$$\mathcal{L}_{\text{quality}} = \text{BCE}(w, w^*),$$

while samples marked as uncertain are ignored. Through this supervision, the gate learns to detect mismatched content, outdated appearance, or enlarged spatial discrepancy, and to output lower reliability in these cases.

Behavior in Imperfect Geographic Conditions. The Reliability Estimation Gate ensures that the overall framework does not become overly dependent on geographic priors. When retrievals are missing, incomplete, or incorrect, the gate attenuates their contribution, allowing the model to fall back smoothly on onboard sensing. The degradation studies in the main paper demonstrate that even as the proportion of missing or perturbed street-view inputs increases, performance degrades gracefully without catastrophic collapse. This stability highlights the gate's ability to maintain balanced fusion and robust perception across real-world variations in street view density, recency, or alignment.

E. Task Definition and Implementation Details

E.1. 3D Object Detection

Task Definition. 3D object detection in autonomous driving aims to localize and classify surrounding traffic participants in an ego-centric 3D coordinate system using multi-view camera inputs. Given a set of synchronized images, the model estimate the position, dimensions, orientation, and semantic category of each object in the scene. The output is typically represented as a collection of oriented 3D bounding boxes, each parameterized by a 3D center, box extents, heading angle, and class label. On nuScenes, performance is commonly quantified using mean Average Precision (mAP) and the nuScenes Detection Score (NDS), which jointly evaluate detection quality across localization, orientation, and related attributes.

We consider BEVDet [48] and BEVFormer [168] as baselines, which are representative multi-view BEV-based detectors on nuScenes [6]. In our setting, geographic information is injected into their BEV feature representations via the Spatial Retrieval Adapter, while the detection heads, training objectives, and evaluation protocol follow the original works. Hyperparameters are detailed in Table 1.

Evaluation Metrics. Metrics are adopted from the official nuScenes detection challenge [6].

- **mAP (↑):** Mean Average Precision, calculated based on 2D center distance on the BEV plane.
- **NDS (↑):** The nuScenes Detection Score, a weighted sum of mAP and five mean True Positive (mTP) metrics: average translation (ATE), scale (ASE), orientation (AOE), attribute (AAE), and velocity (AVE) errors.

$$\text{NDS} = \frac{1}{10}\left[5 \cdot \text{mAP} + \sum_{\text{mTP}\in\text{TPs}} (1 - \min(1, \text{mTP}))\right]$$

E.2. Online Mapping

Task Definition. Online mapping focuses on reconstructing local high-definition map elements that describe the static road topology needed for navigation and downstream decision making. The task takes multi-view camera images as input and predicts structured representations of static road components in an ego-centric bird's-eye-view coordinate frame. Typical targets include lane dividers, lane boundaries, and crosswalks, which are represented as vectorized geometric primitives. The resulting map elements support lane-level reasoning and provide a compact description of the underlying road layout.

The output of this task consists of sets of polylines or curves with associated semantic categories. Because these structures are thin, elongated, and topologically constrained, online mapping is particularly sensitive to occlusion, adverse illumination, and limited sensor range. The lack of explicit depth measurements further increases the difficulty of accurately recovering road geometry from images.

We adopt MapTR [83] and MapTRv2 [85] as baselines for this task. We conduct experiments on the nuScenes [6] dataset. Both methods operate on BEV representations derived from multi-view images and are widely used for vectorized map prediction on nuScenes. In our experiments, geographic images aligned with the ego trajectory are encoded and fused into the BEV features through the Spatial Retrieval Adapter. The prediction heads and training pipelines of the baselines are kept unchanged. Hyperparameters are detailed in Table 1.

Evaluation Metrics. Metrics are based on the MapTR protocol for vectorized map element construction on the nuScenes dataset.

- **AP (↑):** Average Precision is used to evaluate map construction quality. Unlike standard object detection which uses IoU, MapTR utilizes Chamfer distance ($D_{Chamfer}$) to determine whether a predicted map element matches the ground truth. The final AP is calculated by averaging the APs obtained under several matching thresholds $\tau \in T$, where $T = \{0.5m, 1.0m, 1.5m\}$.

$$\text{AP} = \frac{1}{|T|}\sum_{\tau\in T} \text{AP}_\tau$$

- **mAP (↑):** The mean Average Precision is the average of the calculated APs over the three map element classes: pedestrian crossing, lane divider, and road boundary ($N_{cls} = 3$).

$$\text{mAP} = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} \text{AP}_i$$

E.3. Occupancy Prediction

Task Definition. 3D occupancy prediction formulates scene understanding as a volumetric classification problem, where the environment around the ego vehicle is discretized into a 3D grid and each voxel is assigned an occupancy state and a semantic label. The task aims to capture both static infrastructure and dynamic participants in a dense and geometrically explicit representation. This volumetric view of the scene is particularly useful for downstream planning and simulation, as it provides fine-grained information about which regions of space are free, traversable, or blocked.

The model takes multi-view camera images as input, aggregates them into a BEV feature representation, and decodes a multi-class 3D occupancy grid over a fixed spatial range. The Occ3D-nuScenes benchmark [130] reports mean Intersection-over-Union (mIoU) over semantic classes, including driveable area, other flat, sidewalk, terrain, manmade structures, vegetation, and dynamic categories. The task is challenging because large portions of the 3D volume may be unobserved or only partially visible, requiring the model to infer plausible completions based on structural priors and multi-view consistency.

We use FB-OCC [80] and FlashOCC [159] as baselines, which are representative BEV-based occupancy prediction models evaluated on Occ3D-nuScenes [130] in our experiments. Both methods lift multi-view features into a BEV space and decode them into voxel-wise predictions. In our spatial retrieval setting, geographic images provide additional information about static background layout and are fused with the BEV features through the Spatial Retrieval Adapter. The occupancy heads and loss functions remain identical to the original implementations. Hyperparameters are detailed in Table 1.

Evaluation Metrics. Metrics follow the Occ3D-nuScenes benchmark for voxel-level semantic prediction.

- **mIoU (↑):** The mean Intersection-over-Union (IoU) over N_{cls} classes, where TP, FP, and FN are the counts of true positive, false positive, and false negative voxels for a given class.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \quad \text{mIoU} = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} \text{IoU}_i$$

E.4. End-to-End Planning

Task Definition. End-to-end autonomous driving planning aims to generate a safe and kinematically feasible future trajectory for the ego-vehicle directly from raw sensor inputs, conditioned on high-level navigation commands (e.g., turn left, turn right, or go straight). Unlike modular pipelines that rely on intermediate perception outputs like bounding boxes or HD maps, end-to-end approaches optimize the trajectory generation jointly. The model takes multi-view camera images as input to encode the scene context and outputs a sequence of future waypoints $\mathcal{T}_{pred} = \{p_t\}_{t=1}^{T}$ representing the ego-vehicle's planned path over a future horizon T.

In our experiments, we evaluate the planning performance on the nuScenes [6] dataset using VAD [55] as the baseline, which is a representative vectorized planning framework. VAD explicitly models the driving scene using vectorized agent motions and map elements to impose planning constraints, ensuring safety and compliance. In our spatial retrieval setting, we augment the BEV encoder of VAD by adding additional cross attention between the BEV feature and the offline retrieved geographic features through the Spatial Retrieval Adapter. These geographic priors serve as a stable environmental reference to refine the planning trajectory. Hyperparameters are detailed in Table 1.

Evaluation Metrics. Following the specific implementation of ST-P3 [42] and VAD [55], we evaluate the open-loop planning performance using the Average L2 Displacement Error and the Average Collision Rate over different time horizons ($H \in \{1s, 2s, 3s\}$).

- **L2 Displacement Error (L2 ↓):** This metric reports the Average Displacement Error (ADE) accumulated over the time horizon H. It is calculated as the mean Euclidean distance between the predicted waypoints and ground truth waypoints:

$$\text{L2}_H = \frac{1}{N_H} \sum_{t=1}^{N_H} \|\hat{p}_t - p_t^{GT}\|_2$$

where N_H is the number of time steps within the horizon H.

- **Collision Rate (Collision % ↓):** This metric evaluates the safety of the planned trajectory by calculating the frequency of collisions across time steps. Importantly, to account for data noise, a collision at time step t is only penalized if the expert (ground truth) trajectory is collision-free at that moment. The metric is defined as the average percentage of time steps involving a valid collision:

$$\text{Collision}_H = \frac{1}{N_H} \sum_{t=1}^{N_H} \mathbb{I}\left(\mathcal{C}(\hat{p}_t, \mathcal{O}_t) \wedge \neg \mathcal{C}(p_t^{GT}, \mathcal{O}_t)\right) \times 100\%$$

where $\mathcal{C}(p, \mathcal{O})$ indicates whether the ego-vehicle polygon at pose p intersects with the occupancy grid $\mathcal{O}$ of other agents, and $\mathbb{I}$ is the indicator function.

E.5. Generative World Model

Task Definition. Generative world model for autonomous driving aims to synthesize realistic, controllable multi-view driving videos conditioned on structured scene information. Given a short history of surround-view camera frames together with ego-vehicle actions, agent configurations, and environment descriptors, the model learns a conditional distribution over future video sequences. The output is a set of temporally coherent RGB streams for all cameras, which should remain consistent with the specified ego motion, dynamic-agent behaviors, and high-level scene attributes. Following prior work on generative world models [31, 32, 158], performance is typically assessed using distribution-level video metrics such as Fréchet Inception Distance for visual fidelity, complemented by task-specific controllability measures that compare generated content against the conditioning signals (e.g., action or 3D box consistency).

We adopt Unimlvg [14] and MagicDriveDit [31] as a representative multi-view latent diffusion world model baseline. These models are built on video foundation models, where Unimlvg trained to generate 19-frame videos and Magicdrivedit generate 17-frame videos. During testing, Unimlvg uses the first 3 frames as reference frames, and after the first round of generation, the last 3 frames of the generated 19-frame clip are taken as reference for the next round. In total, it performs two rounds of autoregressive generation to produce 35-frame videos for evaluation. Magicdrivedit instead uses only the first frame as the reference frame and directly generates 17-frame videos.

Based on the nuScenes dataset, Unimlvg generates 150 35-frame video clips for testing, whereas Magicdrivedit, under our setup, is evaluated on approximately 2,400 video clips with a frame interval of 15. For each generated clip, we allocate a geographic image to assist generation according to the position of its first and last frames. The detailed training parameters are listed in Table 1.

Evaluation Metrics. Metrics assess the distributional similarity between generated and real videos.

- **FVD (↓) / FID (↓):** Fréchet Video/Inception Distance between the distributions of real-world data (r) and generated data (g), which are modeled as multivariate Gaussians with mean μ and covariance Σ.

$$\text{FVD/FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}\right)$$

Table 1. **Training Hyperparameters for All Baselines.** We report the total batch size across all GPUs. Configurations are aligned with the original papers for a fair evaluation. UVG and MDD are generative models, and their settings reflect the video generation task. * indicates models fine-tuned on the corresponding baseline.

Hyperparameter	MapTR	MapTRv2	FB-OCC	Flash-OCC	BEVDet	BEVFormer	VAD	UVG	MDD
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate	6e-4	6e-4	2e-4	1e-4	2e-6	2e-6	2e-4	8e-5	8e-5
LR Scheduler	CosineAnnealing	CosineAnnealing	LinearWarmup	LinearWarmup	StepDecay	CosineAnnealing	CosineAnnealing	None	LinearWarmup
Weight Decay	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Batch Size (Total)	32	32	32	32	16	2	8	8	32
Training Epochs	110	110	20	24	4*	4*	60	4*	4*
GPUs Used	8 × NVIDIA RTX 4090 (48GB)							8 × NVIDIA H800 (80GB)	

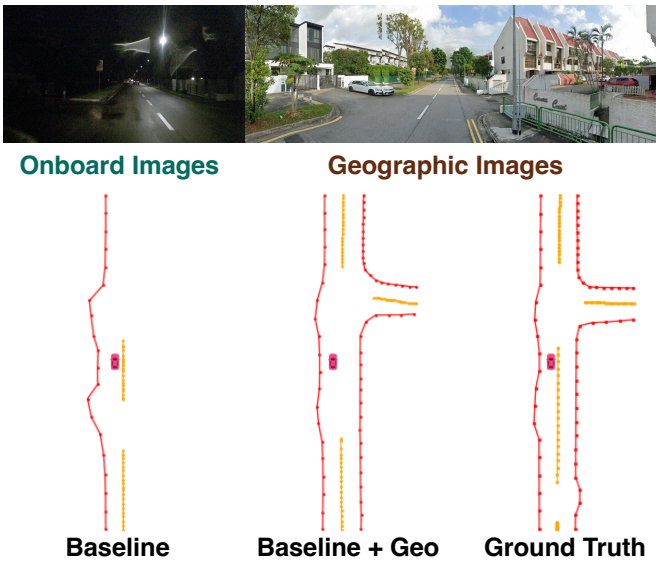

Figure 8. **Online Mapping:** The baseline model(MapTRv2) fails to detect the right-side lane line and intersection due to the dim lighting. Our Spatial Augmented model recovers the crucial road topology by leveraging the stable, bright static environment prior provided by the geographic images, closely matching the Ground Truth.

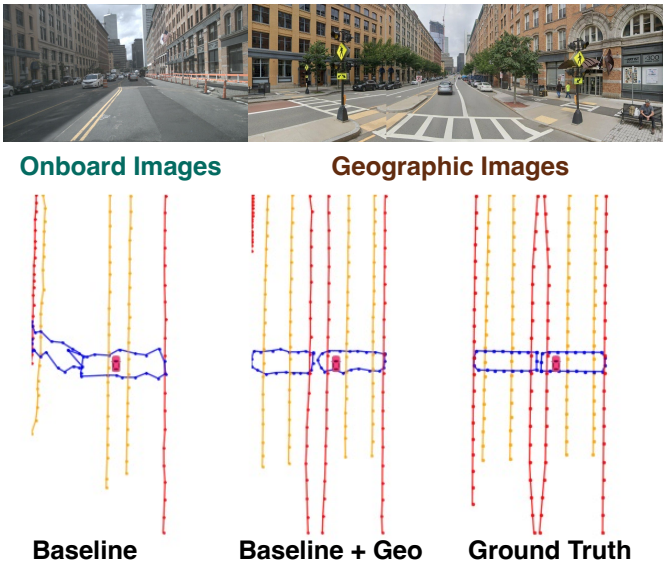

Figure 9. **Online Mapping:** The baseline model struggles to accurately map the pedestrian crossing (blue polygon) near the ego vehicle due to the limited field of view of onboard cameras. The non-ego-centric perspective from the geographic images provides a stable, complete spatial context, enabling our Spatial Augmented model to recover the correct geometric shape, resulting in a prediction highly consistent with the Ground Truth.

F. Extra Qualitative Results

We provide additional qualitative examples to demonstrate the effectiveness of our approach in challenging scenarios. The online mapping results are shown in Fig. 8 with Fig. 9 ,the occupancy prediction results are shown in Fig. 11, the planning results are shown in Fig. 10 and the generation results in Fig. 12.

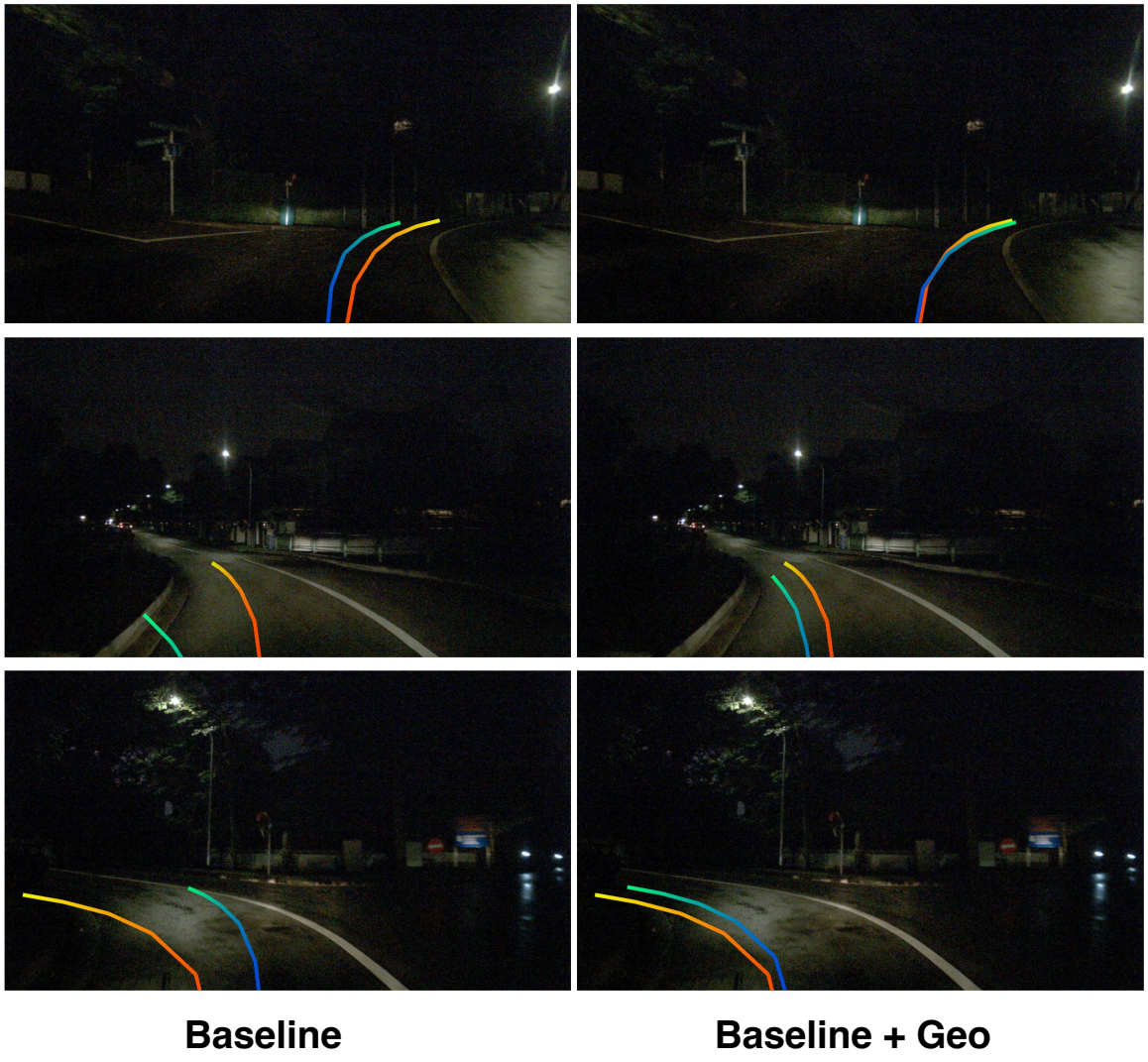

Figure 10. **Planning:** The figure compares the planning performance between the baseline(VAD) and our Spatial Augmented method in night scenes. While the baseline struggles driving in low light, our method acts as a robust guide, generating safer and more consistent trajectories.

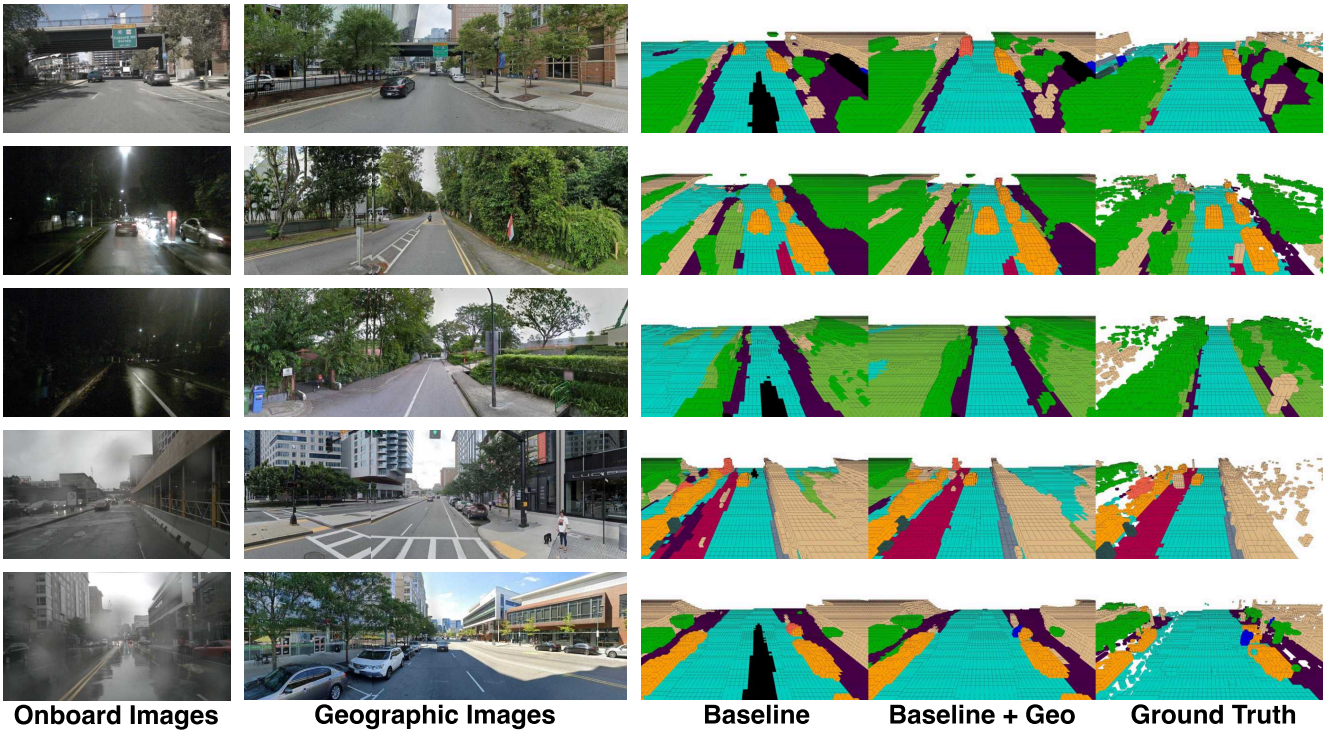

Figure 11. **Occupancy Prediction.** Our method, benefiting from the spatial priors provided by geographic images, produces more complete and coherent occupancy estimations, particularly under challenging conditions such as adverse weather, motion blur, and occlusions.

Figure 12. **Generative World Model.** In the presence of geographic images, the autoregressive generation becomes more stable.

References

[1] Occupancy networks. https://www.thinkautonomous.ai/blog/occupancy-networks/. Accessed July 25, 2024.

[2] Asam openscenario domain-specific language, 2.1.0, 2024. Online.

[3] M. Althoff, M. Koschi, and S. Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *IEEE Intelligent Vehicle Symposium (IV)*, pages 719–726, 2017.

[4] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[7] Tianhui Cai, Yifan Liu, Zewei Zhou, Haoxuan Ma, Seth Z Zhao, Zhiwen Wu, and Jiaqi Ma. Driving with regulation: Interpretable decision-making for autonomous vehicles with retrieval-augmented reasoning via llm. *arXiv preprint arXiv:2410.04759*, 2024.

[8] C. Chang, D. Cao, L. Chen, K. Su, K. Su, Y. Su, F.-Y. Wang, J. Wang, P. Wang, J. Wei, G. Wu, X. Wu, H. Xu, N. Zheng, and L. Li. Metascenario: A framework for driving scenario data description, storage and indexing. *IEEE Transactions on Intelligent Vehicles*, 8(2):1156–1175, 2023.

[9] C. Chang, S. Wang, J. Zhang, J. Ge, and L. Li. Llmscenario: Large language model driven scenario generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(11):6581–6594, 2024.

[10] C. Chang, J. Zhang, J. Ge, Z. Zhang, J. Wei, L. Li, and F.-Y. Wang. Vistascenario: Interaction scenario engineering for vehicles with intelligent systems for transport automation. *IEEE Transactions on Intelligent Vehicles*, 2024.

[11] Jiacheng Chen, Ruizhi Deng, and Yasutaka Furukawa. Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. *arXiv preprint arXiv:2306.01461*, 2023.

[12] Jiacheng Chen, Yuefan Wu, Jiaqi Tan, Hang Ma, and Yasutaka Furukawa. Maptracker: Tracking with strided memory fusion for consistent vector hd mapping. *arXiv preprint arXiv:2403.15951*, 2024.

[13] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023.

[14] Rui Chen, Zehuan Wu, Yichen Liu, Yuxin Guo, Jingcheng Ni, Haifeng Xia, and Siyu Xia. Unimlvg: Unified framework for multi-view long video generation with comprehensive control capabilities for autonomous driving. *arXiv preprint arXiv:2412.04842*, 2024.

[15] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. *arXiv preprint arXiv:2206.04584*, 2022.

[16] Shaoyu Chen, Yunchi Zhang, Bencheng Liao, Jiafeng Xie, Tianheng Cheng, Wei Sui, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene. *arXiv preprint arXiv:2304.09807*, 2023.

[17] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[18] Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. *arXiv preprint arXiv:2412.18607*, 2024.

[19] Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint arXiv:2404.14327*, 2024.

[20] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (11):12878–12895, 2022.

[21] comma.ai. commavq. https://github.com/commaai/commavq, 2023.

[22] Xingyuan Dai, Chao Guo, Yun Tang, Haichuan Li, Yutong Wang, Jun Huang, Yonglin Tian, Xin Xia, Yisheng Lv, and Fei-Yue Wang. Vistarag: Toward safe and trustworthy autonomous driving through retrieval-augmented generation. *IEEE Transactions on Intelligent Vehicles*, 2024.

[23] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning (CoRL)*, 2023.

[24] Wenjie Ding, Limeng Qiao, Xi Qiu, and Chi Zhang. Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3682, 2023.

[25] Jingliang Duan, Yiting Kong, Chunxuan Jiao, Yang Guan, Shengbo Eben Li, Chen Chen, Bingbing Nie, and Keqiang Li. Distributional soft actor-critic for decision-making in on-ramp merge scenarios. *Automotive Innovation*, 7(3):403–417, 2024.

[26] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[27] A. Hogan et al. Knowledge graphs, 2020.

[28] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner, 2018.

[29] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature Communications*, 12(1):748, 2021.

[30] Ruiyuan Gao, Kai Chen, Zhihao Li, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024.

[31] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrivedit: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024.

[32] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024.

[33] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*, 2024.

[34] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2025.

[35] J. Ge, J. Zhang, C. Chang, Y. Zhang, D. Yao, and L. Li. Task-driven controllable scenario generation framework based on aog. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):6186–6199, 2024.

[36] J. Guo, C. Chang, Z. Li, and L. Li. Mixing left and right-hand driving data in a hierarchical framework with llm generation. *IEEE Robotics and Automation Letters*, 9(10):8290–8297, 2024.

[37] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*, 2023.

[38] F. Hauer, I. Gerostathopoulos, T. Schmidt, and A. Pretschner. Clustering traffic scenarios using mental models as little as possible. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1007–1012, 2020.

[39] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 251–257. IEEE, 2020.

[40] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future instance segmentation in bird's-eye view from surround monocular cameras. In *ICCV*, 2021.

[41] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *Technical Report arXiv:2309.17080*, 2023.

[42] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022.

[43] Y. Hu, W. Zhan, and M. Tomizuka. Scenario-transferable semantic graph reasoning for interaction-aware probabilistic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):23212–23230, 2022.

[44] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[45] Yue Hu, Juntong Peng, Sifei Liu, Junhao Ge, Si Liu, and Siheng Chen. Communication-efficient collaborative perception via information filling with codebook. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15481–15490, 2024.

[46] Y. Hu, C. Zhang, B. Wang, J. Zhao, X. Gong, J. Gao, and H. Chen. Noise-tolerant znn-based data-driven iterative learning control for discrete nonaffine nonlinear mimo repetitive systems. *IEEE/CAA Journal of Automatica Sinica*, 11(2):344–361, 2024.

[47] Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. *arXiv:2211.17111*, 2022.

[48] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv:2112.11790*, 2021.

[49] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9223–9232, 2023.

[50] Y. Huang, W. Liu, Y. Li, L. Yang, H. Jiang, Z. Li, and J. Li. Mfe-ssnet: Multi-modal fusion-based end-to-end steering angle and vehicle speed prediction network. *Automotive Innovation*, pages 1–14, 2024.

[51] Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3903–3913, 2023.

[52] Mohamed Manzour Hussien, Angie Nataly Melo, Augusto Luis Ballardini, Carlota Salinas Maldonado, Rubén Izquierdo, and Miguel Ángel Sotelo. Rag-based explainable prediction of road users behaviors for automated driving using knowledge graphs and large language models. *arXiv preprint arXiv:2405.00449*, 2024.

[53] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[54] Bo Jiang, Shaoyu Chen, Xinggang Wang, Bencheng Liao, Tianheng Cheng, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, and Chang Huang. Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. *arXiv preprint arXiv:2212.02181*, 2022.

[55] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[56] Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

[57] Zhou Jiang, Zhenxin Zhu, Pengfei Li, Huan-ang Gao, Tianyuan Yuan, Yongliang Shi, Hang Zhao, and Hao Zhao. P-mapnet: Far-seeing map generator enhanced by both sdmap and hdmap priors. *arXiv preprint arXiv:2403.10521*, 2024.

[58] M Kalfaoglu, Halil Ibrahim Ozturk, Ozsel Kilinc, and Alptekin Temizel. Topomask: Instance-mask-based formulation for the road topology problem via transformer-based architecture. *arXiv preprint arXiv:2306.05419*, 2023.

[59] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, pages 8248–8254. IEEE, 2019.

[60] J. Kerber, S. Wagner, K. Groh, D. Notz, T. Kühbeck, D. Watzenig, and A. Knoll. Clustering of the scenario space for the assessment of automated driving. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 578–583, 2020.

[61] John J. Leonard, Jonathan P. How, Seth J. Teller, Mitch Berger, Stefan Campbell, Gaston A. Fiore, Luke Fletcher, Emilio Frazzoli, Albert S. Huang, Sertac Karaman, Olivier Koch, Yoshiaki Kuwata, David C. Moore, Edwin Olson, Steven C. Peters, Justin Teo, Robert Truax, Matthew R. Walter, David Barrett, Alexander K Epstein, Keoni Maheloni, Katy Moyer, Troy Jones, Ryan Buckley, Matthew E. Antone, Robert Galejs, Siddhartha Krishnamurthy, and Jonathan Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25, 2008.

[62] Bohan Li, Yasheng Sun, Zhujin Liang, Dalong Du, Zhuanghui Zhang, Xiaofeng Wang, Yunnan Wang, Xin Jin, and Wenjun Zeng. Bridging stereo geometry and bev representation with reliable mutual interaction for semantic scene completion. *arXiv preprint arXiv:2303.13959*, 2023.

[63] Bohan Li, Jiajun Deng, Wenyao Zhang, Zhujin Liang, Dalong Du, Xin Jin, and Wenjun Zeng. Hierarchical temporal context learning for camera-based semantic scene completion. In *European Conference on Computer Vision*, pages 131–148. Springer, 2024.

[64] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. *arXiv preprint arXiv:2412.05435*, 2024.

[65] G. Li, W. Zhou, S. Lin, S. Li, and X. Qu. On-ramp merging for highway autonomous driving: An application of a new safety indicator in deep reinforcement learning. *Automotive Innovation*, 6(3):453–465, 2023.

[66] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Enze Xie, Zhiqi Li, Hanming Deng, Hao Tian, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *arXiv preprint arXiv:2209.05324*, 2022.

[67] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, Hao Tian, Enze Xie, Jiangwei Xie, Li Chen, Tianyu Li, Yang Li, Yulu Gao, Xiaosong Jia, Si Liu, Jianping Shi, Dahua Lin, and Yu Qiao. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2151–2170, 2023.

[68] L. Li, N. Zheng, and F.-Y. Wang. A theoretical foundation of intelligence testing and its application for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(10):6297–6306, 2021.

[69] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[70] Tianyu Li, Li Chen, Xiangwei Geng, Huijie Wang, Yang Li, Zhenbo Liu, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, Feng Wen, Ping Luo, Junchi Yan, Wei Zhang, Xiaogang Wang, Yu Qiao, and Hongyang Li. Topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

[71] Tianyu Li, Li Chen, Xiangwei Geng, Huijie Wang, Yang Li, Zhenbo Liu, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, et al. Topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

[72] Tianyu Li, Peijin Jia, Bangjun Wang, Li Chen, KUN JIANG, Junchi Yan, and Hongyang Li. Lanesegnet: Map learning with lane segment perception for autonomous driving. In *The Twelfth International Conference on Learning Representations*, 2024.

[73] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F.-Y. Wang. From features engineering to scenarios engineering for trustworthy ai: I&i, c&c, and v&v. *IEEE Intelligent Systems*, 37(4):18–26, 2022.

[74] Y. Li, C. Tang, S. Peeta, and Y. Wang. Integral-sliding-mode braking control for a connected vehicle platoon: Theory and application. *IEEE Transactions on Industrial Electronics*, 66(6):4618–4628, 2019.

[75] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, 2023.

[76] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9087–9098, 2023.

[77] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. *arXiv preprint arXiv:2406.08481*, 2024.

[78] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. *arXiv preprint arXiv:2504.01941*, 2025.

[79] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025.

[80] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*, 2023.

[81] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

[82] Bencheng Liao, Shaoyu Chen, Bo Jiang, Tianheng Cheng, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction. *arXiv preprint arXiv:2303.08815*, 2023.

[83] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction, 2023.

[84] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*, 2024.

[85] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *International Journal of Computer Vision*, 133(3):1352–1374, 2025.

[86] Si Liu, Chen Gao, Yuan Chen, Xingyu Peng, Xianghao Kong, Kun Wang, Runsheng Xu, Wentao Jiang, Hao Xiang, Jiaqi Ma, et al. Towards vehicle-to-everything autonomous driving: A survey on collaborative perception. *arXiv preprint arXiv:2308.16714*, 2023.

[87] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision (ECCV)*, pages 531–548. Springer, 2022.

[88] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. In *ICCV*, 2023.

[89] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning, 2023.

[90] Zhi Liu, Shaoyu Chen, Xiaojie Guo, Xinggang Wang, Tianheng Cheng, Hongmei Zhu, Qian Zhang, Wenyu Liu, and Yi Zhang. Vision-based uneven bev representation learning with polar rasterization and surface estimation. *arXiv preprint arXiv:2207.01878*, 2022.

[91] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[92] Zihao Liu, Xiaoyu Zhang, Guangwei Liu, Ji Zhao, and Ningyi Xu. Leveraging enhanced queries of point sets for vectorized map construction. *arXiv preprint arXiv:2402.17430*, 2024.

[93] Jia Quan Loh, Xuewen Luo, Fan Ding, Hwa Hui Tew, Junn Yong Loo, Ze Yang Ding, Susilawati Susilawati, and Chee Pin Tan. Cross-domain transfer learning using attention latent features for multi-agent trajectory prediction, 2024.

[94] Enhui Ma, Lijun Zhou, Tao Tang, Zhan Zhang, Dong Han, Junpeng Jiang, Kun Zhan, Peng Jia, Xianpeng Lang, Haiyang Sun, et al. Unleashing generalization of end-to-end autonomous driving with controllable long video generation. *arXiv preprint arXiv:2406.01349*, 2024.

[95] Junyi Ma, Xieyuanli Chen, Jiawei Huang, Jingyi Xu, Zhen Luo, Jintao Xu, Weihao Gu, Rui Ai, and Hesheng Wang. Cam4docc: Benchmark for camera-only 4d occupancy forecasting in autonomous driving applications. *arXiv preprint arXiv:2311.17663*, 2023.

[96] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Y. Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey. *arXiv preprint arXiv:2208.02797*, 2022.

[97] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey. *arXiv preprint arXiv:2208.02797*, 2022.

[98] A. V. Malawade, S. Y. Yu, B. Hsu, H. Kaeley, A. Karra, and M. A. Al Faruque. Roadscene2vec: A tool for extracting and embedding road scene-graphs. *Knowledge-Based Systems*, 242:108245, 2022.

[99] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[100] Jiageng Mao, Boyi Li, Boris Ivanovic, Yuxiao Chen, Yan Wang, Yurong You, Chaowei Xiao, Danfei Xu, Marco Pavone, and Yue Wang. Dreamdrive: Generative 4d scene modeling from street view images. *arXiv preprint arXiv:2501.00601*, 2024.

[101] A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, and F. van Harmelen. An evaluation of knowledge graph embeddings for autonomous driving data: Experience and practice. *Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice*, 2020.

[102] Lili Miao, Shang-Fu Chen, Yu-Ling Hsu, and Kai-Lung Hua. How does c-v2x help autonomous driving to avoid accidents? *Sensors*, 22(2):686, 2022.

[103] Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. *arXiv preprint arXiv:2405.04390*, 2024.

[104] Vandana Narri, Amr Alanwar, Jonas Mårtensson, Christoffer Norén, Laura Dal Col, and Karl Henrik Johansson. Set-membership estimation in shared situational awareness for automated vehicles in occluded scenarios. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 385–392. IEEE, 2021.

[105] P. Nguyen, T. H. Wang, Z. W. Hong, S. Karaman, and D. Rus. Text-to-drive: Diverse driving behavior synthesis via large language models. *arXiv preprint arXiv:2406.04300*, 2024.

[106] Jingcheng Ni, Yuxin Guo, Yichen Liu, Rui Chen, Lewei Lu, and Zehuan Wu. Maskgwm: A generalizable driving world model with video mask reconstruction. *arXiv preprint arXiv:2502.11663*, 2025.

[107] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020.

[108] Cong Pan, Yonghao He, Junran Peng, Qian Zhang, Wei Sui, and Zhaoxiang Zhang. Baeformer: Bi-directional and early interaction transformers for bird's eye view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9590–9599, 2023.

[109] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv preprint arXiv:2309.09502*, 2023.

[110] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[111] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In *ICLR*, 2023.

[112] Liang Peng, Junkai Xu, Haoran Cheng, Zheng Yang, Xiaopei Wu, Wei Qian, Wenxiao Wang, Boxi Wu, and Deng Cai. Learning occupancy for monocular 3d object detection. *arXiv preprint arXiv:2305.15694*, 2023.

[113] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020.

[114] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13218–13228, 2023.

[115] Limeng Qiao, Yongchao Zheng, Peng Zhang, Wenjie Ding, Xi Qiu, Xing Wei, and Chi Zhang. Machmap: End-to-end vectorized solution for compact hd-map construction. *arXiv preprint arXiv:2306.10301*, 2023.

[116] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. Uniformer: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. *arXiv preprint arXiv:2207.08536*, 2022.

[117] Narayanan Elavathur Ranganatha, Hengyuan Zhang, Shashank Venkatramani, Jing-Yan Liao, and Henrik I. Christensen. Semvecnet: Generalizable vector map generation for arbitrary sensor configurations. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2820–2827, 2024.

[118] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021.

[119] Shunli Ren, Siheng Chen, and Wenjun Zhang. Collaborative perception for autonomous driving: Current status and future trend. In *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, pages 682–692. Springer, 2022.

[120] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. In *BMVC*, 2019.

[121] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *European Conference on Computer Vision*, pages 414–430. Springer, 2020.

[122] Oliver Scheel, Luca Bergamini, Maciej Wołczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients, 2021.

[123] Juyeb Shin, Francois Rameau, Hyeonjun Jeong, and Dongsuk Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. *arXiv preprint arXiv:2301.04470*, 2023.

[124] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. 2023.

[125] Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M Alvarez. Centaur: Robust end-to-end autonomous driving with test-time training. *arXiv preprint arXiv:2503.11650*, 2025.

[126] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. *arXiv preprint arXiv:2503.03125*, 2025.

[127] Chen Sun, Ruihe Zhang, Yukun Lu, Yaodong Cui, Zejian Deng, Dongpu Cao, and Amir Khajepour. Toward ensuring safety for autonomous driving perception: standardization progress, research advances, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[128] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

[129] Ardi Tampuu, Tambet Matiisen, Maksym Semikin, Dmytro Fishman, and Naveed Muhammad. A survey of end-to-end driving: Architectures and training methods. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1364–1384, 2020.

[130] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36:64318–64330, 2023.

[131] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023.

[132] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8406–8415, 2023.

[133] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000.

[134] Chris Urmson, Joshua Anhalt, J. Andrew Bagnell, Christopher R. Baker, Robert Bittner, M. N. Clark, John M. Dolan, David Duggins, Tugrul Galatali, Christopher Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matthew McNaughton, Nick Miller, Kevin M. Peterson, Brian Pilnick, Ragunathan Raj Rajkumar, Paul E. Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod M. Snider, Anthony Stentz, William Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, James N. Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25, 2008.

[135] Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 897–904. IEEE, 2022.

[136] Antonin Vobecky, Oriane Siméoni, David Hurych, Spyridon Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. *Advances in Neural Information Processing Systems*, 36, 2024.

[137] Fei Wang, Penglin Dai, Chuzhao Li, Zhangjie Meng, and Kai Liu. Towards communication-efficient collaborative perception: Harnessing channel-spatial attention and knowledge distillation. In *Wireless Artificial Intelligent Computing Systems and Applications*, pages 228–240, Cham, 2025. Springer Nature Switzerland.

[138] J. Wang, A. V. Malawade, J. Zhou, S. Y. Yu, and M. A. Al Faruque. Rs2g: Data-driven scene-graph extraction and embedding for robust autonomous perception and scenario understanding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7493–7502, 2024.

[139] Lening Wang, Wenzhao Zheng, Dalong Du, Yunpeng Zhang, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, Jie Zhou, Jiwen Lu, et al. Stag-1: Towards realistic 4d driving simulation with video generation model. *arXiv preprint arXiv:2412.05280*, 2024.

[140] S. Wang, Y. Zhu, Z. Li, Y. Wang, L. Li, and Z. He. Chatgpt as your vehicle co-pilot: An initial attempt. *IEEE Transactions on Intelligent Vehicles*, 8(12):4706–4721, 2023.

[141] Shuo Wang, Fan Jia, Yingfei Liu, Yucheng Zhao, Zehui Chen, Tiancai Wang, Chi Zhang, Xiangyu Zhang, and Feng Zhao. Stream query denoising for vectorized hd map construction. *arXiv preprint arXiv:2401.09112*, 2024.

[142] W. Wang, A. Ramesh, J. Zhu, J. Li, and D. Zhao. Clustering of driving encounter scenarios using connected vehicle trajectories. *IEEE Transactions on Intelligent Vehicles*, 5(3):485–496, 2020.

[143] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *ECCV*, 2024.

[144] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[145] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[146] Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. *arXiv preprint arXiv:2306.10013*, 2023.

[147] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

[148] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21729–21740, 2023.

[149] Yiming Wu, Ruixiang Li, Zequn Qin, Xinhai Zhao, and Xi Li. Heightformer: Explicit height modeling without extra data for camera-only 3d object detection in bird's eye view. *arXiv:2307.13510*, 2023.

[150] Zhu Xiao, Jinmei Shu, Hongbo Jiang, Geyong Min, Hongyang Chen, and Zhu Han. Overcoming occlusions: Perception task-oriented information sharing in connected and autonomous vehicles. *IEEE Network*, 37(4):224–229, 2023.

[151] Huiyuan Xiong, Jun Shen, Taohong Zhu, and Yuelong Pan. Ean-mapnet: Efficient vectorized hd map construction with anchor neighborhoods. *arXiv preprint arXiv:2402.18278*, 2024.

[152] Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Neural map prior for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17535–17544, 2023.

[153] Zhenhua Xu, Kenneth KY Wong, and Hengshuang Zhao. Insightmapper: A closer look at inner-instance information for vectorized high-definition mapping. *arXiv preprint arXiv:2308.08543*, 2023.

[154] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.

[155] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *arXiv preprint arXiv:2211.10439*, 2022.

[156] Xun Yang, Yunyang Shi, Jiping Xing, and Zhiyuan Liu. Autonomous driving under v2x environment: state-of-the-art survey and challenges. *Intelligent Transportation Infrastructure*, 1:liac020, 2022.

[157] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding, 2019.

[158] Junqi You, Xiaosong Jia, Zhiyuan Zhang, Yutao Zhu, and Junchi Yan. Bench2drive-r: Turning real world data into reactive closed-loop autonomous driving benchmark by generative model. *arXiv preprint arXiv:2412.09647*, 2024.

[159] Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023.

[160] Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model. *arXiv preprint arXiv:2402.10828*, 2024.

[161] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7356–7365, 2024.

[162] Tianyuan Yuan, Yucheng Mao, Jiawei Yang, Yicheng Liu, Yue Wang, and Hang Zhao. Presight: Enhancing autonomous vehicle perception with city-scale nerf priors. *arXiv preprint arXiv:2403.09079*, 2024.

[163] Gongjie Zhang, Jiahao Lin, Shuang Wu, Yilin Song, Zhipeng Luo, Yang Xue, Shijian Lu, and Zuoguan Wang. Online map vectorization for autonomous driving: A rasterization perspective. *arXiv preprint arXiv:2306.10502*, 2023.

[164] J. Zhang, C. Chang, Z. He, W. Zhong, D. Yao, S. Li, and L. Li. CAVSim: A microscopic traffic simulator for evaluation of connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(9):10038–10054, 2023.

[165] K. Zhang, C. Chang, W. Zhong, S. Li, Z. Li, and L. Li. A systematic solution of human driving behavior modeling and simulation for automated vehicle studies. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):21944–21958, 2022.

[166] Yunpeng Zhang, Wenzhao Zheng, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. A simple baseline for multi-camera 3d object detection. *arXiv preprint arXiv:2208.10035*, 2022.

[167] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022.

[168] Li Zhiqi, Wang Wenhai, Li Hongyang, Xie Enze, Sima Chonghao, Lu Tong, Yu Qiao, and Dai Jifeng. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022.

[169] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13760–13769, 2022.

[170] Qiu Zhou, Jinming Cao, Hanchao Leng, Yifang Yin, Yu Kun, and Roger Zimmermann. Sogdet: Semantic-occupancy guided multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7668–7676, 2024.

[171] M. Zipfl, M. Jarosch, and J. M. Zöllner. Traffic scene similarity: A graph-based contrastive learning approach. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 221–227, 2023.