Memorize What Matters: Emergent Scene Decomposition from Multitraverse

Yiming Li[1,2] **Zehong Wang**[1] **Yue Wang**[2,3] **Zhiding Yu**[2]
Zan Gojcic[2] **Marco Pavone**[2,4] **Chen Feng**[1] **Jose M. Alvarez**[2]
[1]NYU [2]NVIDIA [3]USC [4]Stanford University

{yimingli,cfeng}@nyu.edu
{yuewang,zhidingy,zgojcic,mpavone,josea}@nvidia.com

Abstract

Humans naturally retain memories of permanent elements, while ephemeral moments often slip through the cracks of memory. This selective retention is crucial for robotic perception, localization, and mapping. To endow robots with this capability, we introduce `3D Gaussian Mapping (3DGM)`, a *self-supervised*, *camera-only* offline mapping framework grounded in 3D Gaussian Splatting. `3DGM` converts multitraverse RGB videos from the same region into a Gaussian-based environmental map while concurrently performing 2D ephemeral object segmentation. Our key observation is that the environment remains consistent across traversals, while objects frequently change. This allows us to exploit self-supervision from repeated traversals to achieve environment-object decomposition. More specifically, `3DGM` formulates multitraverse environmental mapping as a robust differentiable rendering problem, treating pixels of the environment and objects as inliers and outliers, respectively. Using robust feature distillation, feature residuals mining, and robust optimization, `3DGM` jointly performs 3D mapping and 2D segmentation without human intervention. We build the Mapverse benchmark, sourced from the Ithaca365 and nuPlan datasets, to evaluate our method in unsupervised 2D segmentation, 3D reconstruction, and neural rendering. Extensive results verify the effectiveness and potential of our method for self-driving and robotics.

1 Introduction

Vision-based 3D mapping is essential for autonomous driving but faces two key challenges: *(1)* dynamic objects disrupting multi-view consistency and *(2)* reconstructing accurate 3D structures from 2D images. Existing methods rely on pretrained segmentation models to filter out dynamic objects and LiDARs for better geometry. However, these approaches are hindered by the necessity of human annotations for training, as well as the high costs and limited portability of LiDARs.

Motivated by the aforementioned challenges, we aim to develop a *self-supervised* and *camera-only* 3D mapping approach, reducing the need for human annotations and LiDARs. We consider a practical *multitraverse* driving scenario, where autonomous vehicles repeatedly traverse the same routes or regions at different times. During each traversal, the ego-vehicle encounters new pedestrians and vehicles, similar to how humans navigate the same 3D environment but encounter different groups of passersby each day. Inspired by humans' ability to memorize the **permanent** and ignore the **ephemeral**[1] during repeated spatial navigation, we pose the following question:

[1] We will use *ephemeral* and *transient* interchangeably to refer to objects that temporarily appear or disappear across various traversals of the same location, such as pedestrians, vehicles, or other temporary elements.

Project page: https://3d-gaussian-mapping.github.io

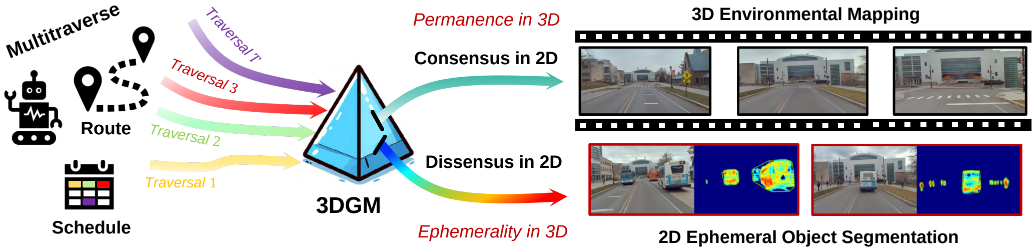


Figure 1: **A high-level diagram of** `3D Gaussian Mapping (3DGM)`**.** Given multitraverse RGB videos, `3DGM` outputs a Gaussian-based environment map (`EnvGS`) and 2D ephemerality segmentation (`EmerSeg`) for the input images. Note that the proposed framework is LiDAR-free and self-supervised.

> *Is it possible to develop an autonomous mapping system that can identify and memorize only the consistent environmental structures of the 3D world across multiple traversals, without relying on human supervision?*

We provide an affirmative answer to this question. Our key insight is using the consensus across repeated traversals as the self-supervision signal, ensuring that the learned map retains only consensus structures (*permanent environment*) while forgetting dissensus elements (*transient objects*). We ground our insight in 3D Gaussian Splatting (3DGS) [1], which models a 3D scene using a group of 3D Gaussians with learnable attributes such as position, color, and opacity. This scene representation provides both geometric and photometric information, benefiting various downstream applications in autonomous driving. We utilize abundant images from multiple traversals to facilitate Gaussian initialization with Structure from Motion (SfM) [2], without using LiDARs. Subsequently, we learn the environmental Gaussians from multitraverse RGB videos by optimizing over the 2D image space.

To optimize a consistent 3D representation from input images with time-varying structures, we treat *multitraverse environmental mapping* as a *robust differentiable rendering* problem where pixels from transient objects are considered outliers. More specifically, we distill self-supervised robust features, denoised DINOv2 [3, 4], into Gaussians to facilitate outlier identification. Afterward, we use a novel feature residual mining strategy to fully exploit the spatial information within the rendering loss map. This strategy aids in precise outlier grouping, enhancing transient object segmentation. Finally, we apply a robust loss function to optimize the 3D environmental Gaussians. Consequently, we can accurately learn the Gaussian-based environment map from inlier pixels and even generate 2D masks of transient objects for free, as illustrated in Fig. 1.

We build the ***Map**ping and segmentation through multitra**verse*** (**Mapverse**) benchmark, sourced from the Ithaca365 [5] and nuPlan [6] datasets to evaluate our method in three tasks: unsupervised 2D segmentation, 3D reconstruction, and neural rendering. Quantitative and qualitative results demonstrate the effectiveness of our method in autonomous driving scenarios.

To summarize, our key innovations are listed as follows.

- **Problem formulation** We address the multitraverse RGB mapping problem through robust differentiable rendering, treating pixels of the environment as inliers and objects as outliers.
- **Technical design** We introduce feature residuals mining to leverage spatial information from rendering loss maps, enabling more accurate outlier segmentation in self-driving scenes.
- **System integration** We build 3D Gaussian Mapping (`3DGM`) that jointly generates 3D environmental Gaussians and 2D ephemerality masks without LiDARs and human supervision.
- **Dataset curation** We build a large-scale multitraverse driving benchmark from real-world datasets for the community, featuring 40 locations, each with no less than 10 traversals, totaling 467 driving video clips and 35,304 images. This dataset will be released for further research.

2 Related Works

Multitraverse driving A vehicle generally operates within the same geographical area, leading to multiple traversals of the same location. This repetition enriches the vehicle's memory of specific places, enhancing its capabilities in perception and localization [7–10]. Regarding perception, the

Hindsight framework [11] utilizes past LiDAR point clouds to learn memory features that are easy to query, thereby addressing the challenges of point sparsity and boosting 3D detection performance. Other studies have leveraged the persistence prior score [12, 13], which quantifies the consistency of a single LiDAR point across multiple traversals, for self-training of detectors and domain adaptation. In localization, a significant number of works focus on either metric [14, 15] or topological [16, 17] localization, aiming to match a query image with a set of reference images collected from different traversals under varying seasonal or lighting conditions. Closely related to our work is [18], which employs multiple traversals to map out ephemeral regions, enhancing monocular visual odometry in dense traffic conditions. However, this approach also depends on the consistency of LiDAR point clouds across traversals, remarking an unexplored gap in leveraging consensus in the 2D image space.

NeRF and 3DGS NeRF has recently revolutionized novel-view synthesis and scene reconstruction with image or video input, boasting a wide range of applications in graphics, vision, and robotics. NeRF employs a volumetric representation and trains neural networks to model density and color. The success of NeRF has sparked a surge in follow-up methods aiming to enhance quality [19–21] and increase speed [22–24]. The recent 3D Gaussian Splatting (3DGS) [1] uses an explicit Gaussian-based representation and splatting-based rasterization [25] to project anisotropic 3D Gaussians onto a 2D screen. It determines the pixel's color by performing depth sorting and α-blending on the projected 2D Gaussians, thus avoiding the complex sampling strategy of ray marching and achieving real-time rendering. Subsequent works have applied 3DGS to scene editing [26], dynamic scene modeling [27, 28], sparse view reconstruction [29], mesh reconstruction [30], semantic understanding [31, 32], and indoor SLAM [33].

NeRF and 3DGS for self-driving Beyond their use in object-centric scenarios and bounded indoor environments, NeRF and 3DGS have also been explored in unbounded driving scenes [34, 35]. Several works address the implicit surface reconstruction of static scenes [36–38]. A large body of research focuses on dynamic scene reconstruction from a single driving log. Most works use a compositional method and rely on bounding annotations/trained detectors to model dynamic objects [39–45]. EmerNeRF [46] is the first self-supervised method to learn 4D neural representations of driving scenes from LiDAR-camera recordings. It couples static, dynamic, and flow fields [24] and leverages the flow field to aggregate multi-frame information to enhance the feature representation of dynamic objects. Another line of research investigates the scalability of the neural representation to model large-scale scenes [47–52]. Block-NeRF [47] segments the scene into separately trained NeRF models, processing camera images from multiple drives, and applies a semantic segmentation model [53] to exclude common movable objects. SUDS takes the input of multitraverse driving logs, leveraging RGB images, LiDAR point clouds, DINO [54], and 2D optical flow [55] for dynamic scene decomposition. In this work, we create an environment map represented by 3DGS without requiring LiDARs, leveraging the multitraverse consensus for self-supervised object removal.

Scene decomposition Traditional background subtraction approaches [56, 57] distinguish moving objects from static scenes by comparing successive video frames and identifying significant differences as foreground elements. Representative works include low-rank decomposition, which treats moving objects in the scene as pixel-wise sparse outliers [58, 59]. These methods are typically used in surveillance applications and are limited to static cameras. Follow-up works [60, 61] investigate background subtraction for mobile robotics, yet suffering from low performance. NeRF has recently emerged as a popular scene representation and has been applied to the self-supervised dynamic-static decomposition of indoor scenes by modeling *time-varying* and *time-independent* components separately [62, 63]. EmerNeRF [46] extends similar intuition to autonomous driving and obtains scene flow for free while achieving dynamic-static decomposition of a single traversal. Yet it still depends on the LiDAR inputs. In this study, we leverage signals of consensus and dissensus across multiple traversals to accomplish *permanence-ephemerality* decomposition using only image inputs.

Vision foundation models Inspired by the success of scaling in NLP [64], the field of computer vision intensively studies large-scale self-supervised pre-training with Transformers [65]. Vision Transformers (ViTs) [66], pre-trained on extensive datasets, achieve excellent image recognition results. DINO [54] further amplifies feature representation capabilities by harnessing self-supervised learning alongside knowledge distillation. Meanwhile, scene layouts emerge within the attention maps, enabling unsupervised semantic understanding. DINOv2 [4] scales up both the data and model size, achieving more robust visual features. Subsequent research focuses on examining noise artifacts to further enhance the performance of self-supervised descriptors, including training-free denoising of ViTs [3] and retraining ViTs with registered tokens [67]. In this work, we leverage denoised DINOv2 features [3,4] to facilitate consensus verification across multiple traversals in pixel space, *as the high-dimensional features prove more resilient to changes in environmental appearance.*

3 3DGS: 3D Gaussian Splatting

3D Gaussian Splatting [1] represents the 3D environment with a set of anisotropic 3D Gaussians, denoted by $\mathbf{G} = \{\mathbf{G}_i \mid i = 1, \ldots, N\}$, where N is the total number of Gaussians. Each Gaussian, $\mathbf{G}_i$, is parameterized by its mean vector $\boldsymbol{\mu}_i \in \mathbb{R}^3$, indicating the position, and a covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times 3}$, defining its shape. To guarantee positive semi-definiteness, the covariance matrix $\boldsymbol{\Sigma}_i$ is further decomposed as $\boldsymbol{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{R}_i^\top$, with $\mathbf{R}_i$ being an orthogonal rotation matrix and $\mathbf{S}_i$ a diagonal scaling matrix. These are stored compactly as a rotation quaternion $\mathbf{q}_i \in \mathbb{R}^4$ and a scaling factor $\mathbf{s}_i \in \mathbb{R}^3$. Each Gaussian also incorporates an opacity value $\alpha_i \in \mathbb{R}$ and a spherical harmonics coefficients $\boldsymbol{\beta}_i$. Therefore, the learnable parameters for the i-th Gaussian are $\mathbf{G}_i = [\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, \alpha_i, \boldsymbol{\beta}_i]$. Rendering from a viewpoint computes the color at pixel $\mathbf{p}$ (denoted by $\mathbf{c}_\mathbf{p}$) via volumetric rendering, integrating K ordered Gaussians $\{\mathbf{G}_k \mid k = 1, \ldots, K\}$ overlapping pixel $\mathbf{p}$, *i.e.*, $\mathbf{c}_\mathbf{p} = \sum_{k=1}^{K} \mathbf{c}_k \alpha_k \prod_{j=1}^{k-1}(1-\alpha_j)$. Here, α_k is derived by evaluating a 2D Gaussian projection [25] from $\mathbf{G}_k$ onto pixel $\mathbf{p}$, multiplied by the Gaussian's learned opacity, and $\mathbf{c}_k$ is the color obtained by evaluating the spherical harmonics of $\mathbf{G}_k$. The Gaussians are sorted by their depth from the viewpoint. The overall objective is to minimize the rendering loss:

$$\mathcal{L} = \sum_t \mathcal{L}_{rgb}(\mathbf{I}_t(\boldsymbol{\xi}_t; \mathbf{G}), \mathbf{I}_t) \tag{1}$$

where $\mathbf{I}_t(\boldsymbol{\xi}_t; \mathbf{G}) \in \mathbb{R}^{w\times h\times 3}$ is the RGB image indexed by t, with spatial dimensions $w \times h$ and rendered from the pose $\boldsymbol{\xi}_t \in \mathfrak{se}(3)$, given Gaussians $\mathbf{G}$. $\mathbf{I}_t \in \mathbb{R}^{w\times h\times 3}$ is the paired ground truth image. $\mathcal{L}_{rgb}$ is a loss function such as L1 loss. Initialized by COLMAP [2], all attributes of $\mathbf{G}$ are learned by executing this view reconstruction task. Meanwhile, adaptive densification and pruning strategies are proposed to improve the fitting of the 3D scene.

4 3DGM: 3D Gaussian Mapping

4.1 Problem Formulation

Assumption We make reasonable assumptions about the stability of the environment and the transience of objects within it. Specifically, we assume that there are no major environmental changes, a realistic expectation when data is collected over a certain period under consistent weather and lighting conditions. Meanwhile, we consider all movable objects to be transient; despite their potential static nature during a particular traversal, they are expected to eventually move somewhere else, allowing the camera to capture dissensus over time.

Setup We conduct offline mapping of a specified spatial area by repeatedly traversing it with vehicles equipped with a monocular camera. The 3D environment map is represented by a set of 3D Gaussians, denoted as $\mathbf{G} = \{\mathbf{G}_i \mid i = 1, \ldots, N\}$. Each $\mathbf{G}_i$ has a set of learnable parameters $[\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, \alpha_i, \boldsymbol{\beta}_i, \mathbf{f}_i]$, where $\mathbf{f}_i \in \mathbb{R}^d$ is a self-supervised d-dimensional semantic feature such as DINO [4] for a more robust representation, and other parameters follow 3DGS as detailed in Sec. 3. This mapping approach not only captures the geometry but also the photometry of the environment, yielding a comprehensive scene representation for downstream tasks such as geometry reconstruction and view synthesis. The input to our approach comes from a set of unposed images, sourced from multitraverse RGB videos, denoted by $\mathbf{I} = \{\mathbf{I}_t \in \mathbb{R}^{w\times h\times 3} \mid t = 1, \ldots, T\}$, where T is the total number of images, w and h are the width and height of each image, respectively.

Target The target is to refine $\mathbf{G}$ to a level where it can accurately render images $\mathbf{I}_t(\boldsymbol{\xi}_t; \mathbf{G})$ that closely match the real images $\mathbf{I}_t$, captured from specific poses $\boldsymbol{\xi}_t$. Although $\mathbf{G}$ represents a 3D spatial map, the input images encompass 4D information with both spatial and temporal dimensions. Hence, our method needs to adeptly differentiate between the environment and ephemeral objects, like pedestrians and vehicles, to maintain robustness against pixels that represent transient entities. This

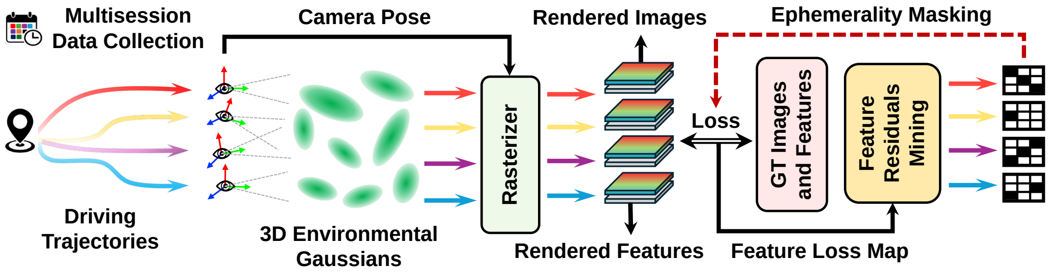


Figure 2: **An overall illustration of 3DGM.** Given RGB camera observations collected at different times, we use COLMAP to obtain the camera poses and initial Gaussian points. Then we utilize splatting-based rasterization to render both RGB images and robust features from the environmental Gaussians. We further leverage feature residuals to extract the object masks by mining spatial information of the residuals. Finally, we utilize the ephemerality masks to finetune the 3D Gaussians.

necessitates addressing a robust estimation problem, where the outliers are transient objects—those that are either in motion or capable of moving—while the inliers are backgrounds.

4.2 Overall Architecture

Figure 2 shows our overall workflow. Given RGB images $\mathbf{I}$ collected across multiple traversals, we first leverage the classic Structure from Motion (SfM) [2] to jointly reconstruct sparse points for the initialization of Gaussians and obtain the camera poses $\boldsymbol{\xi} = \{\boldsymbol{\xi}_t \mid t = 1, \ldots, T\}$. We then utilize the differential rendering pipeline of 3DGS to learn the positions, rotations, scales, opacities, colors, and semantic features of the 3D environmental Gaussians $\mathbf{G}$, supervised by ground truth RGB $\mathbf{I}$ and self-supervised feature maps [4] denoted by $\mathbf{F} = \{\mathbf{F}_t \in \mathbb{R}^{w \times h \times d} \mid t = 1, \ldots, T\}$. Then we exploit the feature residual maps to extract ephemeral object masks denoted by $\mathbf{M} = \{\mathbf{M}_t \in \mathbb{R}^{w \times h} \mid t = 1, \ldots, T\}$. Finally, we finetune 3D Gaussians $\mathbf{G}$ through robust differentiable rendering by leveraging the ephemerality masks. In summary, 3DGM includes the three stages denoted by Initialization, EmerSeg, and EnvGS, as shown in Appendix A.1. We detail each stage from Sec. 4.3 to 4.5.

4.3 Initialization: Structure from Motion

The SfM pipeline frequently faces challenges in single-traversal scenarios, largely due to the limited scene coverage achieved with RGB observations collected along a narrow and long camera trajectory. Conversely, RGB images from multiple traversals offer a broader array of viewpoints, significantly improving the triangulation and bundle adjustment processes. Additionally, this approach can leverage the 2D consensus of hand-crafted features in the correspondence search, providing inherent robustness against transient objects, which manifest as dissensus pixels across traversals. Moreover, our empirical experiments underscore the importance of the number of traversals for smooth initialization. A reduction in traversals can lead to a lack of sufficient image data, thereby failing the SfM initialization.

4.4 EmerSeg: Emerged Ephemerality Segmentation by Feature Residuals Mining

Feature distillation We utilize robust feature representations to enhance consensus verification, as the feature space exhibits better robustness against lighting variations and embodies semantic meanings, facilitating the decomposition of the transient objects by removing groups of semantically dissensus pixels. We minimize the following RGB and feature rendering loss:

$$\mathcal{L} = \sum_t (\mathcal{L}_{rgb}(\mathbf{I}_t(\boldsymbol{\xi}_t; \mathbf{G}), \mathbf{I}_t) + \mathcal{L}_{feat}(\mathbf{F}_t(\boldsymbol{\xi}_t; \mathbf{G}), \mathbf{F}_t)) \tag{2}$$

where $\mathbf{I}_t(\boldsymbol{\xi}_t; \mathbf{G}) \in \mathbb{R}^{w \times h \times 3}$ and $\mathbf{F}_t(\boldsymbol{\xi}_t; \mathbf{G}) \in \mathbb{R}^{w \times h \times d}$ are the rendered RGB image and feature map given pose $\boldsymbol{\xi}_t \in \mathfrak{se}(3)$ and Gaussians $\mathbf{G}$. $\mathbf{I}_t$ and $\mathbf{F}_t$ are the corresponding ground truth RGB and feature map. $\mathcal{L}_{rgb}$ and $\mathcal{L}_{feat}$ are loss functions for RGB images and semantic features. *As inlier pixels substantially outweigh outlier pixels, the model is primarily steered by gradients from consensus inlier pixels towards learning permanent features. As a result, pixels manifesting high loss in feature space are very likely to be outliers.*

Feature residuals mining We derive transient object masks by leveraging the spatial information in the feature residual maps, as shown in the right column of Fig. XVI~XXI. After training, we normalize the feature residuals and suppress pixels with residual values below a predefined threshold. Contours are then extracted from the normalized residual maps using spatial gradient information [68]. We refine these contours by applying spatial priors to eliminate those that are too small or located in the sky. Finally, we merge nearby contours and extract a convex hull for each merged contour. Ultimately, ephemerality masks $\mathbf{M}$ are produced from simple postprocessing of feature residuals without additional training. More details are shown in Appendix A.2.

4.5 `EnvGS`: Environmental Gaussian Splatting via Robust Optimization

After obtaining ephemerality masks $\mathbf{M}$, we focus on minimizing the following robust loss function (taking L1 loss as an example):

$$\mathcal{L} = \sum_t \mathcal{L}_{rgb}(\mathbf{M}_t \odot \mathbf{I}_t(\boldsymbol{\xi}_t; \mathbf{G}), \mathbf{M}_t \odot \mathbf{I}_t) \tag{3}$$

where $\mathbf{M}_t$ is an ephemerality mask for the t-th image to downgrade the influence of outlier pixels. Optionally, we employ a depth smoothness loss and sky masks to further improve the geometry reconstruction, as illustrated in Appendix A.3.

4.6 Comparison to Arts

The pioneering work addressing similar problems is NeRF-W [69], which learns volumetric representations from unconstrained photo collections. It employs uncertainty estimation to mask transient objects situated in image areas of high uncertainty. The following research efforts propose to learn a transient mask, aiming to eliminate occluders [70, 71]. Another related work is RobustNeRF [72] which models distractors in training data as outliers of an optimization problem and proposes a form of robust estimation for NeRF training.

We have three main differences from prior works.

- **Target problem** We formulate robotic multitraverse RGB mapping as a robust differentiable rendering problem, unlike previous works that focus on object-centric neural rendering of outdoor landmarks or multiple objects in indoor scenarios.
- **Scene decomposition** Our method enables a clearer decomposition of foreground and background, producing both 2D segmentation and 3D environmental Gaussians without any supervision. This represents a significant improvement over previous methods, which produce only blurry results in outdoor scenarios.
- **Technical novelty** We use Gaussian Splatting instead of the conventional NeRF approach. Our robust feature distillation and feature residuals mining fully exploit the spatial information of the rendering loss map, resulting in much better ephemerality segmentation.

5 Experiments

Dataset Most NeRF benchmarks [38, 44, 46] for driving focus on a single-traversal video of the Waymo [73] or nuScenes [74]. To address the gap, we introduce the first *unsupervised **Mapping** and segmentation via multitra**verse*** (**Mapverse**) benchmark, which comprises **Mapverse-Ithaca365** (see Appendix B.1) and **Mapverse-nuPlan** (see Appendix B.2) derived from the Ithaca365 [5] and nuPlan [6] datasets, respectively. **Mapverse** features 40 locations, each with 10∼16 traversals, yielding a total of 467 videos and 35,304 images. Due to space constraints, we present results for **Mapverse-Ithaca365** (20 locations, 200 videos, 20,000 images) in the main text, with additional results in **Mapverse-nuPlan** provided in Appendix F~H. Sample data are visualized in Figs. I~IV.

Task and implementation We benchmark three tasks: *(1) unsupervised 2D ephemerality segmentation*, *(2) 3D reconstruction*, and *(3) neural rendering* in multitraversal driving. Our benchmark can inspire wide applications in unsupervised perception, autolabeling, camera-only 3D reconstruction and neural simulation in self-driving and robotics. For efficiency, we compress feature dimensions from 768 to 64 using PCA. Our model uses KL divergence for feature alignment and L1 loss for RGB reconstruction. All experiments are conducted on a single NVIDIA RTX 3090 GPU.

Table 1: **Mean IoU of unsupervised vs. five supervised methods in Mapverse-Ithaca365.** * indicates the model without training on our dataset.

Unsup. Sup.	EmerSeg (Ours)	STEGO* [82]	STEGO [82]	CAUSE [83]
PSPNet [75]	**42.22**	20.55	22.22	19.12
SegViT [76]	**46.16**	21.18	23.57	19.64
InternImage [79]	**46.34**	21.29	23.68	19.93
Mask2Former [77]	**42.28**	20.83	23.03	20.88
SegFormer [78]	**45.14**	21.31	23.78	20.63

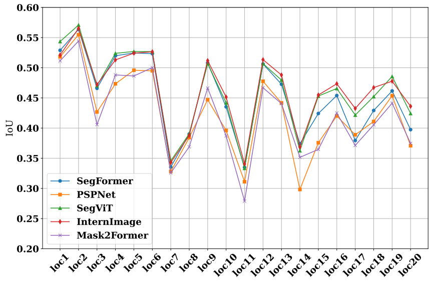


Figure 3: **IoU at 20 locations in Ithaca, NY.**

5.1 Unsupervised 2D Ephemeral Object Segmentation

Task setup Our EmerSeg can segment ephemeral traffic participants in a multitraverse image collection, *without any supervision*. This will help identify moving objects like vehicles and pedestrians, as well as static objects with the potential for movement, such as parked cars or traffic cones. We use a *training-as-optimization* pipeline and adopt the *Intersection over Union (IoU) metric* for evaluation. Regarding comparison methods, we employ several state-of-the-art *semantic segmentation* models trained with human annotations to create pseudo ground-truth masks for transient objects (pedestrians, vehicles, bicyclists, and motorcyclists). We also compare EmerSeg with *unsupervised segmentation* methods. We report the main comparison results in Sec. 5.1.1 and ablation studies in Sec. 5.1.2.

5.1.1 Quantitative and Qualitative Evaluations

Comparison against supervised methods We compare our method with state-of-the-art (SOTA) semantic segmentation methods: PSPNet [75], SegViT [76], Mask2Former [77], SegFormer [78], and InternImage [79]. *Note that these methods require dense pixel-level annotations to learn semantics.* We directly use these models trained on either ADE20K [80] or Cityscapes [81] to produce masks on **Mapverse-Ithaca365**. The overall IoU scores of EmerSeg average around 0.45 compared to the five supervised models; see Tab. 1. IoU scores across 20 locations are detailed in Fig. 3, with seven locations surpassing 50% IoU, and the highest score reaching 56% compared to SegFormer. These results highlight the promising potential of our unsupervised segmentation paradigm.

Comparison against unsupervised methods We compare EmerSeg with two SOTA unsupervised segmentation methods, *i.e.*, STEGO [82] and CAUSE [83]. We train both methods on our dataset using their unsupervised objectives. *Note that these unsupervised baseline methods cannot grasp the semantics or the concept of ephemerality and can only perform clustering within a single image.* Following prior work, we use a Hungarian matching algorithm to align the unlabeled clusters with pseudo ground-truth masks for evaluation. As shown in Tab. 1, EmerSeg significantly outperforms STEGO and CAUSE, with a 21.36-point (89.8%) IoU improvement over STEGO using SegFormer masks. More importantly, EmerSeg can understand ephemerality, a capability lacking in prior works.

Qualitative comparison EmerSeg performs well in various lighting and weather conditions, effectively segmenting cars, buses, and pedestrians; see Fig. 4. However, it struggles with small or distant objects due to low feature map resolution. *We empirically find that small objects have minimal impact on neural rendering as they occupy few pixels.* Additional qualitative results are in Fig. V, with visualizations of baseline methods in Fig. VI. Detailed limitations are discussed in Appendix I.

Computation time Figure VII illustrates the convergence of our segmentation method, showing a rapid increase in IoU during the initial iterations, which stabilizes around iteration 4,000. Notably, a resolution of 110×180 requires only 2,000 iterations to achieve an IoU score exceeding 40%, taking ∼8 minutes on a single NVIDIA RTX 3090 GPU for 1,000 images from 10 traversals of a location.

5.1.2 Ablation Studies

Segmentation performance benefits from more traversals We evaluate 2D segmentation on 100 images from a single traversal, using inputs from varying numbers of traversals; see Tab.2. Starting at 15.15% with one traversal, the IoU jumps to 42.31% with two, and continues to rise: 53.16% at 8

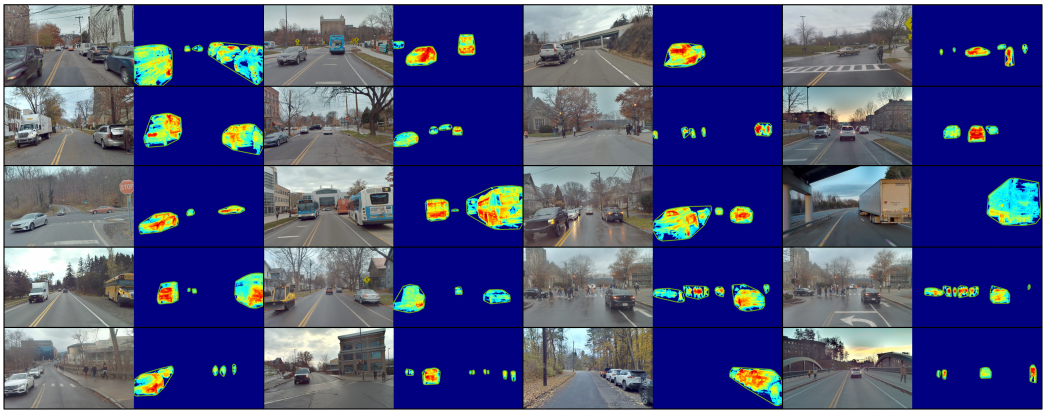


Figure 4: **Qualitative evaluations of `EmerSeg` in Mapverse-Ithaca365.**

Table 2: **Ablation Study Results of `EmerSeg` in Mapverse-Ithaca365.**

Number of Traversals		Feature Dimension			Feature Resolution			Feature Backbone	
#	**IoU (%)**	**Dim.**	**Runtime**	**IoU (%)**	**Res.**	**Size (MB)**	**IoU (%)**	**Backbone**	**IoU (%)**
1	15.15	4	00:13:50	9.45	25×40	0.3	28.61	DINO [54]	16.51
2	42.31	8	00:16:50	10.91	50×80	1.0	35.91	Denoised DINO [3]	14.95
3	46.62	16	00:18:37	26.32	70×110	1.9	40.09	DINOv2 [4]	35.14
5	53.68	32	00:24:48	37.51	110×180	5.0	**44.13**	Denoised DINOv2 [3]	**44.13**
9	54.50	64	00:40:25	**44.13**	140×210	7.4	42.48	DINOv2-reg [67]	23.51
10	**56.01**	128	01:13:53	42.55	160×260	10.5	41.19	Denoised DINOv2-reg [3]	36.30

and 56.01% at 10 traversals. This shows a clear trend of improving IoU with more traversals, with significant gains between 1 and 2. Detailed visualizations are in Fig. VIII.

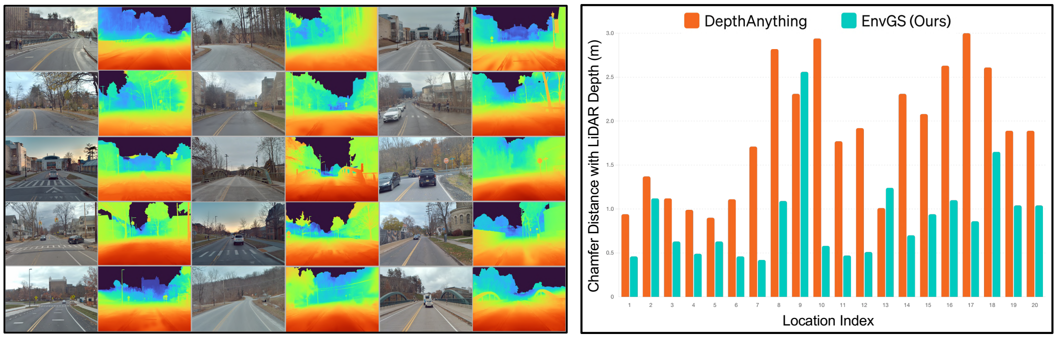
Effective segmentation requires 32 feature dimensions We use PCA to compress the dimensions of DINOv2 features to save computation and storage. Our tests on segmentation performance at various dimensions revealed that 32 is an approximate threshold; IoU scores decrease significantly to around 10%-25% when the number of dimensions falls below 32, as shown in Tab. 2. Qualitative comparisons of different feature dimensions are demonstrated in Fig. IX.

A resolution of 70×110 can achieve an IoU >40% Table 2 shows IoU at various feature resolutions and sizes. IoU improves significantly as resolution increases from 25×40 (28.61%, 0.3 MB) to 110×180 (44.13%, 5.0 MB). However, higher resolutions like 140×210 and 160×260 result in slightly lower IoU scores of 42.48% and 41.19%, despite larger sizes. This indicates an optimal resolution at 110×180, balancing accuracy and efficiency. Visualizations at different resolutions are in Fig. X.

Vision foundation model matters in unsupervised segmentation We use robust features from self-supervised vision foundation models like DINO [54], DINOv2 [4], and DINOv2 with registers [67]. Additionally, we employ DVT [3] to reduce grid-like artifacts in ViT feature maps. As shown in Tab. 2, Denoised DINOv2 outperforms other models, highlighting the importance of robust, discriminative features for identifying transient clusters. Detailed visualizations are in Fig. XI.

5.2 3D Environment Reconstruction

Task setup Our `EnvGS` can extract 3D points from Gaussian Splatting, enabling the reconstruction of 3D environments from camera-only input while effectively ignoring transient objects across repeated traversals. We utilize a *training-as-optimization* pipeline and employ the *Chamfer Distance (CD) metric* for quantitative evaluation. For our comparison baseline, we use the state-of-the-art DepthAnything [84] model, which is trained with a combination of LiDAR ground truth (GT) depth data and unlabeled image data. This approach ensures that DepthAnything leverages diverse data sources to achieve satisfactory performance in zero-shot depth estimation.



(a) Depth Visualizations

(b) Quantitative Comparison across locations

Figure 5: **Qualitative and quantitative evaluation of 3D geometry in Mapverse-Ithaca365.**

Quantitative results Figure 5 demonstrates the large reduction in Chamfer Distance (CD) achieved by `EnvGS` across nearly all locations. Our method achieves an average CD of approximately 0.9 meters, showcasing its precision in 3D reconstruction. Notably, there are five locations where the CD is even lower than 0.5 meters, highlighting the good accuracy of our approach in these areas. In contrast, DepthAnything has an average CD of around 1.9 meters, indicating a notable performance gap between the two methods. More importantly, our method avoids the need for costly LiDAR sensors during training, making it a cost-effective autonomous mapping solution for self-driving and robotics. Leveraging techniques such as mesh reconstruction [30] and 2D Gaussian Splatting [85] could further enhance the geometric reconstruction capabilities of our method.

Qualitative results Figure 5 showcases depth visualizations of `EnvGS` across various driving scenarios. The depth maps generated by `EnvGS` exhibit superior accuracy, with smooth transitions from near to far objects and well-defined edges of scene structures. Additionally, `EnvGS` effectively removes transient objects without human supervision. Visualizations in 3D are shown in Fig. XII.

5.3 Neural Environment Rendering

Task setup Our `EnvGS` can also achieve novel view synthesis through splatting-based rasterization. The challenge lies in ensuring the environment rendering automatically bypasses the non-environment pixels, *i.e.*, transient objects. We evaluate the quality of rendered images using three metrics: Learned Perceptual Image Patch Similarity (LPIPS), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR). Given the absence of ground truth RGB images for clean backgrounds, we utilize the pretrained SegFormer [78] model to isolate foreground regions, allowing us to focus our evaluation exclusively on the quality of the background rendering.

Baseline methods Our baseline methods include two NeRF-based methods, leveraging the implementation framework of iNGP [24]. The first, VanillaNeRF, constructs the scene within a single, static hash table and directly learns grid features from multitraverse images. In contrast, RobustNeRF [72] introduces an adaptive weighting mechanism to filter out outliers. In addition to the original 3DGS framework, we introduce two 3DGS-based baseline methods. 3DGS+RobustNeRF integrates the loss function from RobustNeRF, and 3DGS+SegFormer utilizes masks generated by a supervised segmentation model. For a fair comparison, all methods exclusively rely on camera images as input.

Results and discussions Table 3 presents a quantitative comparison of various methods, showing that 3DGS-based approaches outperform NeRF-based methods. Adding the RobustNeRF loss function does not improve rendering quality in driving scenes. However, incorporating SegFormer or EmerSeg masks achieves the best LPIPS and SSIM. This is notable within a purely self-supervised framework, showcasing the potential of our self-supervised paradigm in pushing the boundaries of neural mapping. We present qualitative examples in Fig. 6, where it is evident that the original 3DGS model struggles with accurately reconstructing background regions affected by transient objects. More interestingly, our method can identify and mask out not only the objects themselves but also their associated non-environmental elements, such as shadows, as shown in the third and sixth columns of Fig. 6. More qualitative examples can be found in Fig. XIV.

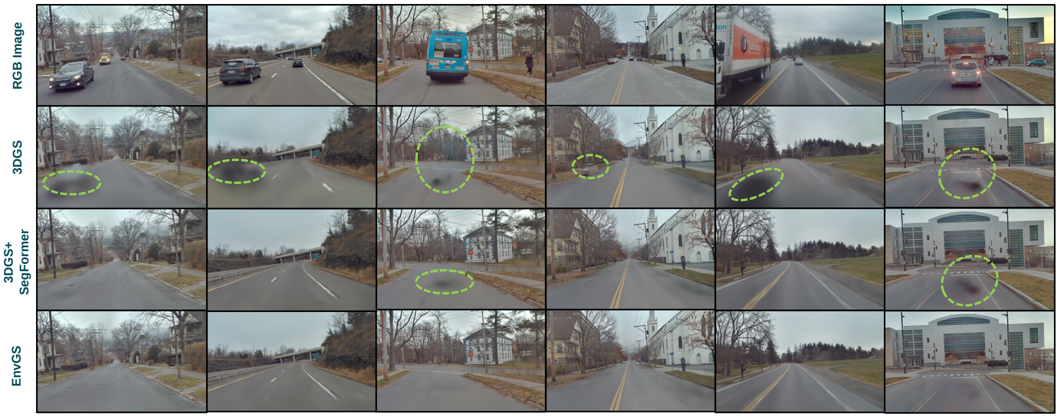


Figure 6: **Qualitative evaluations of the environment rendering.** Our method demonstrates robust performance against transient objects, and can even outperform the method equipped with a pretrained model in some cases. Notably, this includes the effective removal of object shadows.

Table 3: **Quantitative evaluation of novel view synthesis.** We set test/training views as 1/8. Pixels corresponding to transient objects are removed in the evaluations since we do not have ground truth background pixels in these regions occluded by transient objects.

Metrics \ Methods	**VanillaNeRF** [24]	**RobustNeRF** [72]	**3DGS+RobustNeRF**	**3DGS** [1]	**3DGS+SegFormer**	**EnvGS (Ours)**
LPIPS (↓)	0.423	0.443	0.416	0.227	0.212	0.213
SSIM (↑)	0.603	0.609	0.654	0.798	0.806	0.806
PSNR (↑)	19.18	19.22	19.97	22.92	22.81	22.78

6 Conclusion

Broader impacts The concept of vision-only neural representation learning through repeated traversals extends beyond object segmentation and environment mapping, benefiting the vision and robotics communities. With a neural map prior, our approach becomes a powerful self-supervised framework for change detection and object discovery. This capability to render and analyze multitraverse environments over time is crucial for identifying environmental changes, aiding in early intervention for deforestation, urban expansion, or post-disaster assessments. Additionally, our method can serve as a baseline for autolabeling 2D masks and has potential for 3D autolabeling with LiDAR integration.

Limitations Our method faces limitations in modeling large environmental variations, including nighttime conditions, major seasonal shifts, and adversarial weathers. We also note the presence of noise in the segmentation outputs caused by motion blur or appearance shifts. Leveraging temporal information or more powerful vision foundation models could help address this issue. More discussions can be found in Appendix I.

Summary We introduce `3D Gaussian Mapping (3DGM)`, a novel self-supervised, camera-only framework that utilizes repeated traversals for simultaneous 3D environment mapping (`EnvGS`) and 2D unsupervised object segmentation (`EmerSeg`). Additionally, we develop the **Mapverse** benchmark, comprising nearly 500 driving video clips from the Ithaca365 and nuPlan datasets. Our method's effectiveness in unsupervised 2D segmentation, 3D reconstruction, and neural rendering is validated through both qualitative and quantitative assessments in repeated driving scenarios. Furthermore, `3DGM` opens new research opportunities, such as online unsupervised object discovery and offline autolabeling. We believe our work will advance vision-centric and learning-based self-driving and robotics, setting new standards in multitraverse setups and self-supervised scene understanding.

Acknowledgement

We express our deep gratitude to Jiawei Yang and Sanja Fidler for their valuable feedback throughout the project. We also thank Yurong You and Carlos A. Diaz-Ruiz for their support with the Ithaca365 dataset, and Shijie Zhou for his help with high-dimensional feature rendering in 3DGS. Yiming Li gratefully acknowledges support from the NVIDIA Graduate Fellowship Program.