Supplementary Material: Neural Atlas Graphs for Dynamic Scene Decomposition and Editing

Jan Philipp Schneider[1,2] **Pratik Singh Bisht**[1] **Ilya Chugunov**[2]

Andreas Kolb[1] **Michael Moeller**[1,3] **Felix Heide**[2]

[1]University of Siegen [2]Princeton University [3] Lamarr Institute

This supplementary document provides further method details, including specific aspects of the camera model and our coarse-to-fine training scheme. We also expand on description of our dataset and experimental design, included ablation studies, provide additional quantitative results, visual examples, and edits across both autonomous driving and challenging outdoor video sequences. We recommend reviewing the accompanying video, which provides a compelling summary of our key visual contributions.

Given the different modalities of our supplementary materials, we provide the high-resolution videos within a google drive folder and the code within a github repository along this document for further details.

To give an overview of the provided videos within the google drive folder, we briefly highlight the structure:

- `overview.mp4` - contains our overview video, showcasing our key visual results and comparisons. For detailed examples see below.
- `edits/[manuscript|supplement]/figure_[Number]` - contains the videos matching the given figure number in either our manuscript or this supplementary.
- `visuals/[waymo|davis]/[sequence]` - contains all reconstruction for Waymo [24] and Davis [19], and also decompositions for the latter one. We choose the abbreviation ORE for OmniRe [4], ERF for EmerNeRF [31], LNA for Layered Neural Atlases [10], ORF for OmnimatteRF [13] and GT for the ground truth videos.

To provide a better overview of the remaining supplementary material, we provide a table of contents.

Table of Contents

A Additional Method Details

A.1 Camera Model

For rendering and scene interaction, we require a mapping from image coordinates to a 3D world reference system. We utilize the standard pinhole camera model parameterized by intrinsic K and extrinsic $g(t)$ matrices. Considering the projection of a single pixel (u, v) at timestamp t, the ray origin o and direction d in a world reference system can be computed by:

$$\begin{aligned} \hat{d}(u,v) &= (K^{-1} \odot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot f), & \hat{o}(u,v) &= \hat{d}(u,v) - \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix}, \\ o(u,v,t) &= g(t) \odot \hat{o}(u,v), & d(u,v,t) &= R(t) \odot \hat{d}(u,v) \end{aligned} \tag{1}$$

for a camera projection plane lying at z=0. For better readability, we avoided stating homogeneous vector conversions. The inverse of the intrinsic matrix $K \in \mathbb{R}^{3\times3}$ (2) is used to convert a pixel in the camera's local coordinate system. Further, the extrinsic matrix $g(t) \in \mathbb{R}^{3\times4}$, converts from the camera's local into the world coordinate system. These matrices can be defined as:

$$g(t) = \underbrace{\begin{bmatrix} 1 & -r^z & r^y_i \\ r^z_i & 1 & -r^x_i \\ -r^y_i & r^x_i & 1 \end{bmatrix}}_{R(t)} \underbrace{\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}}_{T(t)}, K^{-1} = \begin{bmatrix} 1/fm_x & 0 & -p_x/fm_x \\ 0 & 1/fm_y & -p_y/fm_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The values comprising the intrinsic matrix K are typically provided by the camera manufacturer, whereby $f \in \mathbb{R}$ defines the focal length, m_x, m_y the image width and height, and p_x, p_y define the principal point. While camera extrinsics are generally provided in autonomous driving datasets, these values are susceptible to inaccuracies due to sensor miscalibration or accumulated odometry drift. Furthermore, when estimated using structure-from-motion or neural methods, such as RoDynRF [14] in our outdoor experiments, the resulting poses may also contain noise. To refine these, we

utilize the same spline-based offset learning approach [5] as discussed for our nodes to map t to its correspondences control points $\mathcal{P}^{\text{T}}_{\text{cam},i}, \mathcal{P}^{\text{R}}_{\text{cam},i}$ using interpolation. The learning process will adjust for possible shifts in the camera rotation $R(t)$ and translation $T(t)$, recalling our definitions (3, 4):

$$\begin{aligned} T(t) &= \tilde{T}_t + \eta_{\text{T}} \cdot S(t,\, \mathcal{P}^{\text{T}}_{\text{cam}}) \\ R(t) &= \tilde{R}_t \cdot q(\eta_{\text{R}} \cdot S(t,\, \mathcal{P}^{\text{R}}_{\text{cam}})) \end{aligned} \tag{3}$$

$$\mathcal{P}^{\text{T}}_{\text{cam}} = \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix}_i \right\}_{i=0}^{P}, \quad \mathcal{P}^{\text{R}}_{\text{cam}} = \left\{ \begin{bmatrix} r^x \\ r^y \\ r^z \end{bmatrix}_i \right\}_{i=0}^{P} \tag{4}$$

whereby $S : [0,1] \times \mathbb{R}^P \to \mathbb{R}^F$ denotes the cubic hermite spline interpolation [6], as discussed in [5], and $\mathcal{P}^{\text{T}}_{\text{cam}}, \mathcal{P}^{\text{R}}_{\text{cam}} \in \mathbb{R}^{P\times 3}$ being zero-initialized learnable translation and rotation offsets of the camera. We further denote the rotation vector to unit quaternion operation as $q : [0, 2\pi)^3 \to \mathbb{H}$ for $\mathbb{H}$ being the set of unit-quaternions.

Given such definition, the number of control points $P \in \mathbb{N}$ can be used to encourage smooth motion, e.g. by setting it smaller than the number of frames F in the video $\mathcal{I}$ ($P = F/2$), or keeping it equal to the number of frames to keep the expressivity. The prior-known positions are stated as $\tilde{T}_t \in \mathbb{R}^{F\times 3}$ and $\tilde{R}_t \in \mathbb{H}^F$ describing camera translation and rotation respectively. To control the influence of the learned offsets with introduced temperature weights $\eta_{\text{T}} = \eta_{\text{R}} = 0.5$.

A.2 Coarse-to-fine Optimization

To limit the expressiveness of the view-dependent fields $\mathcal{F}_{i,\phi}$ to model as few changes as possible, as well as enforcing the planar flow field to firstly learn coarse alignment, we apply a coarse-to-fine learning strategy by masking the hash-encoding using a sparsity function [5] $\text{sparse}(\cdot, \tau)$ based on the training progress $\tau \approx \text{clamp}(0.05 + \sin(\text{epoch} \cdot \pi/1.6 \cdot \text{max\_epoch}))$. For epoch being the current epoch in training and max_epoch = 80 the total epochs to conduct. Empirically, sparse deactivates several encoding dimensions E from the multi-resolution hash encodings $\mathcal{H}_{i,\phi} : [0,1]^4 \to \mathbb{R}^E, \mathcal{H}_{i,f} : [0,1]^2 \to \mathbb{R}^E$ for a node i, setting their activations to zero and activates them when training progresses. Correspondingly, the view- and flow-neural fields $\mathcal{F}_{i,\phi}, \mathcal{F}_{i,f}$ may be rewritten as

$$\mathcal{F}_{i,\phi}(x) = \mathcal{N}_{i,\phi}(\text{sparse}(\mathcal{H}_{i,\phi}(x,\phi), \tau)), \tag{5}$$

$$\mathcal{F}_{i,f}(x) = \mathcal{N}_{i,f}(\text{sparse}(\mathcal{H}_{i,f}(x), \tau)), \tag{6}$$

for $\mathcal{N}_{i,\phi} : \mathbb{R}^E \to \mathbb{R}^4$ being the view-dependent MLP and $\mathcal{N}_{i,f} : \mathbb{R}^E \to \mathbb{R}^{P_f\times 2}$ being the flow MLP, predicting P_f flow control points, correspondingly. Further, $x \in [0,1]^2$ denotes the intersection point in planar coordinates and $\phi \in [0,1]^2$ its normalized spherical view angle. Additionally, the expressiveness of the model and its learnable parameters are controlled using a *phase-based learning strategy*, which we further detail below.

Note: while we denoted the color and opacity of the view-dependent field $\mathcal{F}_{i,\phi}$ in the main manuscript separately to increase readability, e.g. $\mathcal{F}_{i,\phi,c}, \mathcal{F}_{i,\phi,\alpha}$, they share parameters.

A.3 Phase-based Learning

Our training strategy employs a three-phase optimization approach combined with the previously mentioned coarse-to-fine learning strategy to effectively optimize the various components of each node. In the first phase, from epoch 0 to 5, only the positional parameters $\mathcal{P}^{\text{T}}_i, \mathcal{P}^{\text{R}}_i, \mathcal{P}^{\text{T}}_{\text{cam}}, \mathcal{P}^{\text{R}}_{\text{cam}}$ [1] are optimized to compensate for positional errors of the objects and camera. In the second phase, epoch 5 to 20, the color and opacity fields $\mathcal{F}_{i,c}, \mathcal{F}_{i,\alpha}$ are additionally optimized. In the last third phase, starting from epoch 20, all parameters $\mathcal{P}^{\text{T}}_i, \mathcal{P}^{\text{R}}_i, \mathcal{P}^{\text{T}}_{\text{cam}}, \mathcal{P}^{\text{R}}_{\text{cam}}, \mathcal{F}_{i,c}, \mathcal{F}_{i,\alpha}, \mathcal{F}_{i,f}, \mathcal{F}_{i,\phi}$ are optimized together.

[1] Note: $\mathcal{P}^{\text{R}}_i$ is not optimized in our main automotive experiments.

A.4 Parametrization

In the following we detail the parametrization of our atlas nodes. Our model fundamentally distinguishes between two types of information for each atlas node: fixed initial conditions (preconditioning) and the core learnable neural fields.

Fixed Initial Conditions To establish a robust starting point for optimization, we initialize the base color $\tilde{C}_i$ and base alpha $\tilde{A}_i$ for each object's appearance. These non-learnable base textures are derived via an initial forward projection (using the camera model in Sec. A.1) of the masked image within a single reference frame onto the object's position-initialized plane. We use the image corresponding to the mask with the largest size as reference. The position initialization itself is carried out using our initial translation and rotation parameters $\tilde{T}_{i,t}, \tilde{R}_{i,t}$ (3, 4), which are extracted from 3D bounding boxes (if available) or image homographies based on the masked region, combined with a monocular depth estimation. These base parameters remain fixed, while the learned components correct for initialization errors.

Learnable Neural Fields The core appearance and motion are then captured by our learnable neural fields: the color field $\mathcal{F}_{i,c}$, opacity field $\mathcal{F}_{i,\alpha}$, flow field $\mathcal{F}_{i,f}$, and view-dependent field $\mathcal{F}_{i,\phi}$. These fields are explicitly designed and optimized to serve distinct, disentangled roles:

- The color $\mathcal{F}_{i,c}$ and opacity $\mathcal{F}_{i,\alpha}$ fields (5, 6) are primarily responsible for modeling the *view-agnostic, canonical appearance* of the object in its atlas space.
- The flow field $\mathcal{F}_{i,f}$ (7) enables the representation of non-rigid motion by warping the canonical appearance across time, facilitating better editability by maintaining consistent base texture across frames.
- The view-dependent field $\mathcal{F}_{i,\phi}$ (5, 6) is designed to capture subtle view-dependent effects (e.g., specularities, reflections) that cannot be explained by the canonical appearance or flow alone.

The optimization process distinguishes these components through our phase-based (Sec. A.3) and coarse-to-fine (Sec. A.2) learning strategy. By initially limiting the expressiveness of the view-dependent field (via sparse encoding) and progressively activating it, we implicitly regularize the model by prioritizing the non-view-dependent fields ($\mathcal{F}_{i,c}, \mathcal{F}_{i,\alpha}, \mathcal{F}_{i,f}$) for primary appearance and motion capture. This forces the components relevant to editing and flow-mapping to learn the majority of the information, ensuring a disentangled representation where $\mathcal{F}_{i,\phi}$ only incorporates subtle, additional changes.

Number of Parameters We state the parameterization of the learnable components of each Neural Atlas Graphs (NAG) node in Tab. 1, while we refer to Sec. B.3 for a description of each field's architecture. The translation and rotation control points $\mathcal{P}_i^{\mathrm{T}} \in \mathbb{R}^{P\times 3}, \mathcal{P}_i^{\mathrm{R}} \in \mathbb{R}^{P\times 3}$ for each object i are dependent on the scene length F and expected smoothness. The camera consists of the same number of translation and rotation parameters. The background will have no learnable position parameters due to its static definition and has no opacity field $\mathcal{F}_{i,\alpha}$ due to its constant opacity of 1. While the number of parameters may be decreased based on the expected size of an object to increase efficiency, we use a single, unified size for all objects for simplicity.

Table 1: Number of learnable parameters for a single NAG node.

Component		Learnable Parameters
Color Field	$\mathcal{F}_{i,c}$	3,720,488
Flow Field	$\mathcal{F}_{i,f}$	3,720,488
Opacity Field	$\mathcal{F}_{i,\alpha}$	3,720,488
View-Dependent Field	$\mathcal{F}_{i,\phi}$	6,716,320
Translation (single control point)	$\mathcal{P}_i^{\mathrm{T}}$	3
Rotation (single control Point)	$\mathcal{P}_i^{\mathrm{R}}$	3

B Experimental Details

B.1 Baselines

In the following we briefly describe our comparison baselines within the Automotive and Outdoor domain.

Automotive Baselines Within the autonomous driving scenes, we evaluate against OmniRe (ORe) [4], a recent dynamic 3D Gaussian Splatting (3DGS) method, which was explicitly designed for autonomous driving scenes including a dedicated SMPL-based human [15] model for pedestrians and showing peak visual performance on the Waymo dataset [24]. Further, given its object-specific architecture, it allows for positional edits, but lacks support for texture editing. We also compare against EmerNeRF (ERF) [31], a state-of-the-art dynamic scene reconstruction method, which leverages learned dynamics models and neural radiance fields to capture complex object motion and interactions, including non-rigid transformations. Although ERF is not object-specific, it serves as a recent and relevant NeRF baseline.
Note on ORe Scene Decomposition: Although the authors provide visualizations of scene decompositions within their manuscript, we could not find the corresponding implementation in the provided codebase. Therefore, we adapted their evaluation scripts to specify and render individual object IDs for decomposition comparisons, while leaving the core implementation untouched.

Outdoor Baselines Our evaluation for outdoor videos, conducted on the DAVIS dataset [19], compares our approach against both recent texture editing and state-of-the-art video matting methods. Layered Neural Atlases (LNA) [10] serves as our texture editing baseline. LNA operates by learning a 2D coordinate mapping, at test time, that projects pixels from all video frames onto a single texture atlas. This atlas can then be edited directly, with the changes re-projected back to all frames for scene manipulation. OmnimatteRF (ORF) is included as the most recent video matting baseline. These models are designed for robust layer separation, aiming to cover objects and associated effects (such as shadows) by learning a 2D foreground layer per-segmented object in image space, situated on top of a 3D background modeled by a radiance field. Crucially, as the 2D foreground layers are generated on a per-frame basis by a U-Net [21], ORF contains no editable layer representation - all information is encoded within the learned U-Net weights.

For all our evaluations, we use the codebase provided by the respective authors unless explicitly stated. We used the recommended settings by the authors and only changed parameters to ensure a fair comparison (e.g., training on a higher resolution). For specific details on parameter changes, we refer to Sec. B.2.

B.2 Datasets

Driving Scenes This section provides a detailed description of the specific subset of the Waymo Open Dataset [24] used for evaluating our proposed method. As outlined in the main manuscript, we specifically selected scenes characterized by small ego-vehicle movement but a high density of dynamic objects, frequent occlusions, and significant variations in object motion emphasizing editability. Our evaluation was conducted on a total of 7 distinct scene segments, from which we extracted 25 subsequences, ranging from 21 to 89 frames sampled at 10Hz. During the subsequence creation, we excluded frames containing corrupted bounding box annotations, as well as longer sequences where no significant object intersection occurred. The sequence identifiers and ranges are stated in Tab. 2. For the remaining images within these subsequences, we segmented all objects[2] for which bounding box information was available and that exhibited significant motion or caused substantial occlusions. Representative ground truth images from each of these sequences, showcasing the generated instance segmentation masks, are visualized in Fig. 1. For all methods we used all images of the datasets (as per experiment's subset division) in the native resolution (1920×1280) to train the models, yielding a representation of maximal visual expressiveness. Since ORe explicitly removes lens distortion during its preprocessing pipeline, we also apply this undistortion step for our method. We note that, due to the ERF model's implementation, no undistortion process is carried out.

[2]We provide these masks along our code base.

Consequently, all methods are evaluated against their respective ground truth targets (distorted or undistorted) to ensure a fair scene reconstruction comparison.

Table 2: Waymo [24] dataset sequences within our automotive evaluation. We stating the range, as respective inclusive start and end indices, forming our 25 subsequences.

Sequence	Segment Specifier	Range
s-125	segment-12511696717465549299	0 - 40, 40 - 93, 93 - 124, 124 - 149
s-141	segment-14133920963894906769	2 - 53, 53 - 101, 102 - 173, 174 - 197
s-203	segment-2036908808378190283	3 - 58, 60 - 107
s-324	segment-3247914894323111613	0 - 42, 42 - 96, 96 - 161, 161 - 197
s-344	segment-3441838785578020259	0 - 51, 52 - 95, 95 - 135, 135 - 197
s-952	segment-9521653920958139982	0 - 63, 64 - 140, 141 - 198
s-975	segment-9758342966297863572	0 - 68, 69 - 99, 99 - 162, 175 - 195

Outdoor Scenes For evaluating the generalization of our method to diverse outdoor scenarios, we utilized a subset of the high-resolution DAVIS dataset [19], a common benchmark in video object segmentation and matting. Specifically, we selected the same 15 sequences also used by the baseline methods, ORF [13] and LNA [10], ensuring a direct basis for comparison across varied objects, backgrounds, and camera motion. We employed the dataset provided instance masks, and combined them into a single foreground mask per frame due to LNA's implementation. Consistent with ORF, we used RodynRF [14] for initial camera pose estimation. Recognizing that the original evaluation resolutions for ORF (428 x 270) and LNA (768 × 432) were significantly lower than our capabilities, we leveraged more computational resources to evaluate our method and LNA on the full DAVIS resolution (up to 1920 x 1080), with the exception of the *lucia* sequence. Due to LNA's high memory demands on this longer scene, we down-sampled *lucia* to 960 x 540 for LNA only. For ORF, given its original compute limitations and our focus on high-resolution performance, we down-sampled the input images by a factor of two (e.g., to 960 x 540) across all sequences, followed by bilinear interpolation of its output to the original resolution for accurate comparison.

B.3 Neural Atlas Graphs Evaluation

NAG Training Our NAG is trained for 80 epochs, whereby each epoch consists of 2.8×10^8 ray-casts into the scene. Each epoch is subdivided into 140 batches, and each batch consists of 100,000 spatial ray-casts which are simultaneously evaluated along 20 random timestamps [3]. We use the Adam [12] optimizer, with an initial learning rate of 0.001 in combination with a "ReduceLROnPlateau" scheduler, which will be activated from epoch 20 on.

Atlas Node Architecture The neural fields $\mathcal{F}_{i,c}, \mathcal{F}_{i,\alpha}, \mathcal{F}_{i,f}$ and $\mathcal{F}_{i,\phi}$ within every atlas node are parameterized by 5-layer MLPs (64 neurons, ReLU), while the input coordinates are encoded with a 16-level multi-resolution hash encoding [17] (4 features/level, scale 1.61, hashmap size 17, base resolution 4, linear interpolation). We state the actual sizes in a dedicated section A.4. For the Waymo [24] dataset, the spline-based motion model of each node utilizes a number of control points $P = F$ equal to the number of images F in the sequence. This is necessary to capture the potentially rapid camera motion (10Hz sampling), caused by oscillation on ego vehicle stops, which a lower-resolution spline cannot accurately represent. For the DAVIS [19] dataset, we set the number of control points to $P = F/2$, allowing for a smoother representation of the nodes, yielding a slightly more robust approach to compensate for inaccuracies in initialization. The effect of the control points is briefly studied within our ablations Sec. C.4. The training runtime ranges from approximately 2 to 6 hours depending on scene complexity and length, using a machine with a NVIDIA L 40 GPU and 64 GB RAM. Our reproducible code and dataset preparation schemes are available at: `https://github.com/jp-schneider/nag`.

[3] Based on the dynamic architecture of the NAG, leading to a different total parameter sizes, we decrease in populated scenes the number of ray-casts per batch and increase the batches per epoch to fit the model into the available VRAM.

Figure 1: Ground truth references and mask examples out of the studied autonomous driving Waymo sequences [24]. Displayed are sequences in order: s-125, s-141, s-203, s-324, s-344, s-952, s-975. The sequences containing various objects and motion patterns. For our NAG representation, each of the masked instances will be attributed to its own atlas node.

C Results

This supplementary section provides a comprehensive extension of the results presented in the main manuscript, necessitated by space constraints. We begin by recalling our main quantitative results in Sec. C.1, now including inter-frame standard variations to rigorously assess significance and temporal consistency. Following this, we dedicate Sec. C.2 to evaluate editing quality using explicit temporal consistency measures. Subsequently, we analyze our model's performance by detailing the partial limitations of the NAG method under large ego-motion conditions (Sec. C.3). We then proceed to ablate our model, conducting extensive ablation studies in Sec. C.4, where we analyze network sizes, the impact of key components, and sensitivity to input masks. Finally, we provide a rich set of

additional visual results in Sec. C.5, including automotive scene reconstruction comparisons, editing figures demonstrating positional and time shifts, object insertions, and removals. This is followed by further reconstruction, decomposition, and editing results on DAVIS [19] outdoor scenes. Concluding this additional results section, we benchmark against three additional Gaussian Splatting baselines for autonomous driving in Sec. C.6 and provide a transparent overview of measured training times for representative scenes across all methods of our main manuscript in Sec. C.7.

C.1 Additional Quantitative Results

We extend the quantitative analysis from our main manuscript, benchmarking our approach against the recent OmniRe [4] —a 3D 3DGS framework — and EmerNeRF [31] a recent dynamic NeRF model. We now explicitly provide inter-frame standard deviation measures to quantify temporal consistency. Tab. 3 lists the PSNR, SSIM [29], and LPIPS [36] scores for each scene, along with their inter-frame standard deviations over all different sub-segments of individual sequences, measuring temporal consistency.

We further isolate and assess the dynamic elements by partitioning them into a rigid "Vehicle" category and a non-rigid "Human" category. This division allows us to specifically evaluate how each class benefits from our underlying rigid-motion model. Tab. 4 reports per-class PSNR and SSIM [29] results, including accompanying inter-object standard deviations over the sub-segments. The results demonstrate substantial improvements over the strongest baseline, confirming that our gains stem not merely from improved background rendering but from the high fidelity of our model in capturing even non-rigid motion.

Beyond the improvements in overall and object-based quality, the competitively low values for the inter-frame standard deviations highlight the high temporal consistency of our method. Our PSNR score of $\pm$ 0.91 compares favorably against OmniRe ($\pm$ 1.39). While this is slightly worse than the $\pm$ 0.78 achieved by EmerNeRF, EmerNeRF achieves this stability at a significantly lower quality (34.93 dB PSNR versus our 41.85 dB PSNR). For SSIM and LPIPS we improve against both baselines. Also, on object-based consistency, we achieve comparable temporal consistency in PSNR, while improving it on SSIM. This demonstrates our method's capability to learn a high-quality and temporally stable scene representation.

To verify generalization, we test our method on diverse outdoor sequences from the DAVIS dataset [19]—a high-resolution (up to 1920×1080) benchmark commonly used by matting methods [10, 16, 13]. Following the selection of 15 sequences in [10, 13] featuring varied objects, complex backgrounds, and dynamic camera moves, we summarize our results in Tab. 5, including inter-frame standard deviations for PSNR, SSIM [29], and LPIPS [36]. In terms of visual quality, we significantly outperform the baselines in PSNR, SSIM, and LPIPS. While our standard deviations measuring temporal consistency are competitive in SSIM and LPIPS, our PSNR stability is weaker ($\pm$ 1.3 vs. $\pm$ 0.66). Crucially, this is achieved alongside a quality improvement of over 7 dB PSNR. We attribute this weakness to challenging scenes (e.g. tennis) which include highly non-rigid and rapid motion. This type of motion can lead to artifacts due to flow collapse and the difficulty of learning large and rapidly changing flow vectors within our spline-based smooth flow assumption.

To further quantify the quality and temporal consistency of our method against the core baselines, we computed the Fréchet Video Distance (FVD) [25] metric, a dedicated video quality evaluation method originally targeted for generative videos, proposed to better align with human judgment than PSNR or SSIM. Additionally, we evaluated the Temporal (T)-LPIPS, which is applied inter-frame wise to indicate differences in machine perception and can therefore be interpreted as an additional temporal consistency metric. Since inter-frame differences are also induced by changes in the scene or camera, T-LPIPS scores must be interpreted relative to the T-LPIPS of the Ground Truth video. The FVD and T-LPIPS metrics are presented in Tab. 6. For this evaluation, we use the ablation sequences (s-141, s-975) from the Waymo Open Dataset and the sequences (blackswan, bear, and boat) from DAVIS. Our method achieves the best FVD scores across the evaluated sequences, aligning with the perceptual improvements observed in our earlier PSNR, SSIM, and LPIPS evaluations. In terms of temporal consistency (T-LPIPS), our method's score is the closest to that of the Ground Truth (GT) video, indicating a similar degree of fidelity in inter-frame changes.

Table 3: Quantitative Evaluation on Dynamic Driving Sequences of the Waymo [24] Open Driving Dataset. The temporal consistency is measured by the inter-frame standard deviation (± STD), which is calculated over sub-segments and mean-aggregated per sequence. Best results are in bold. ORe refers to OmniRe [4], and ERF to EmerNeRF [31]. Our method compares very favorably in overall quality, showing higher consistency in SSIM and LPIPS over the baselines and highly competitive consistency in PSNR.

Seq.	PSNR ↑			SSIM ↑			LPIPS ↓		
	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF
s-975	**40.21** ±1.11	37.35 ±1.73	34.83 ±1.65	**0.976** ±0.004	0.968 ±0.005	0.937 ±0.013	**0.058** ±0.012	0.080 ±0.005	0.143 ±0.017
s-203	**43.15** ±0.39	36.93 ±1.14	36.07 ±0.43	**0.978** ±0.001	0.966 ±0.002	0.936 ±0.003	**0.070** ±0.004	0.094 ±0.003	0.205 ±0.005
s-125	**43.32** ±0.49	38.74 ±0.87	35.20 ±0.48	**0.980** ±0.003	0.970 ±0.002	0.933 ±0.005	**0.057** ±0.007	0.079 ±0.003	0.182 ±0.006
s-141	**42.55** ±1.60	36.14 ±1.29	34.83 ±0.53	**0.978** ±0.003	0.964 ±0.003	0.924 ±0.006	**0.057** ±0.005	0.087 ±0.005	0.178 ±0.011
s-952	**41.89** ±0.59	39.67 ±0.84	35.32 ±0.84	0.976 ±0.003	**0.977** ±0.003	0.938 ±0.008	0.058 ±0.006	**0.050** ±0.002	0.120 ±0.012
s-324	**40.85** ±1.31	32.58 ±2.21	33.63 ±0.57	**0.977** ±0.002	0.953 ±0.010	0.926 ±0.005	**0.038** ±0.004	0.071 ±0.009	0.124 ±0.007
s-344	**41.84** ±0.52	36.67 ±1.40	35.24 ±0.77	**0.983** ±0.001	0.973 ±0.003	0.946 ±0.006	**0.031** ±0.002	0.043 ±0.003	0.084 ±0.005
Mean	**41.85** ±0.91	36.78 ±1.39	34.93 ±0.78	**0.978** ±0.002	0.967 ±0.004	0.934 ±0.007	**0.051** ±0.006	0.070 ±0.004	0.142 ±0.010

Table 4: Quantitative Evaluation of Human and Vehicle Rendering on Waymo [24] Driving Sequences. The stated standard deviations (± STD), are calculated following Tab. 3, and mean-aggregated per object and sub-sequence.

Seq.	Vehicle PSNR ↑			Vehicle SSIM ↑			Human PSNR ↑			Human SSIM ↑		
	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF
s-975	**46.79** ±1.21	33.09 ±3.37	30.21 ±1.73	**0.991** ±0.001	0.939 ±0.038	0.820 ±0.035	**45.37** ±1.58	32.99 ±2.60	28.53 ±1.13	**0.989** ±0.002	0.927 ±0.021	0.777 ±0.030
s-203	**41.90** ±1.89	30.45 ±3.14	27.10 ±1.89	**0.986** ±0.005	0.910 ±0.046	0.774 ±0.053	**45.40** ±1.65	34.85 ±2.81	33.54 ±1.29	**0.986** ±0.004	0.950 ±0.017	0.901 ±0.016
s-125	**41.00** ±1.90	28.72 ±2.42	24.55 ±1.24	**0.989** ±0.004	0.878 ±0.054	0.709 ±0.049	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
s-141	**43.21** ±1.44	33.22 ±2.05	27.36 ±1.21	**0.981** ±0.007	0.929 ±0.028	0.744 ±0.036	**44.22** ±1.61	33.31 ±2.55	28.86 ±1.67	**0.986** ±0.005	0.907 ±0.044	0.769 ±0.051
s-952	**40.94** ±1.46	31.15 ±2.59	27.70 ±2.23	**0.986** ±0.004	0.928 ±0.036	0.810 ±0.061	**40.45** ±2.82	32.32 ±2.35	28.10 ±2.22	**0.968** ±0.021	0.894 ±0.039	0.740 ±0.065
s-324	**41.71** ±1.56	31.03 ±3.41	27.87 ±1.96	**0.986** ±0.004	0.921 ±0.048	0.798 ±0.053	**44.12** ±1.95	32.09 ±2.63	26.40 ±1.78	**0.988** ±0.005	0.894 ±0.041	0.689 ±0.065
s-344	**43.97** ±1.69	33.02 ±2.29	30.65 ±1.43	**0.985** ±0.007	0.931 ±0.019	0.835 ±0.018	**40.99** ±2.97	30.20 ±2.57	25.94 ±1.40	**0.975** ±0.016	0.882 ±0.045	0.721 ±0.051
Mean	**42.88** ±1.56	31.69 ±2.73	28.09 ±1.67	**0.986** ±0.005	0.922 ±0.037	0.787 ±0.043	**42.94** ±2.21	32.24 ±2.55	27.78 ±1.67	**0.981** ±0.010	0.901 ±0.039	0.744 ±0.052

C.2 Assessment of Editing Quality

Quantifying video quality for edited scenes, particularly within decomposition-based methods like ours, remains an open problem. Notably, related works such as Layered Neural Atlases (LNA) [10] and OmnimatteRF (ORF) [13] also omit a direct quantitative evaluation of edited video quality. We posit that this omission stems from two major hurdles:

Firstly, a direct, quantitative assessment of video quality is complicated by the lack of available ground truth data. This area is currently the subject of intensive research in Blind Video Quality

Table 5: Quantitative evaluation results on the Davis Dataset [19] of diverse outdoor scenes with inter-frame standard deviation ($\pm$ STD) over all images from each scene. The best results are in bold for all metrics. Our method consistently yields higher quality than its competitors OmnimatteRF (ORF) [13] and Layered Neural Atlases (LNA) [10], while maintaining competitive temporal stability. (Best results are in bold.)

Sequence	PSNR ↑			SSIM ↑			LPIPS ↓		
	Ours	ORF	LNA	Ours	ORF	LNA	Ours	ORF	LNA
bear	**33.47** $\pm$1.42	24.88 $\pm$0.52	26.51 $\pm$0.72	**0.934** $\pm$0.027	0.658 $\pm$0.020	0.771 $\pm$0.018	**0.091** $\pm$0.030	0.464 $\pm$0.011	0.287 $\pm$0.015
blackswan	**36.36** $\pm$0.53	26.67 $\pm$0.98	29.26 $\pm$0.48	**0.938** $\pm$0.005	0.739 $\pm$0.031	0.815 $\pm$0.014	**0.097** $\pm$0.010	0.458 $\pm$0.032	0.318 $\pm$0.020
boat	**35.83** $\pm$0.42	28.63 $\pm$0.31	30.15 $\pm$0.48	**0.932** $\pm$0.005	0.761 $\pm$0.012	0.816 $\pm$0.011	**0.099** $\pm$0.009	0.376 $\pm$0.013	0.274 $\pm$0.011
car-shadow	**36.67** $\pm$1.57	29.26 $\pm$0.38	28.47 $\pm$0.48	**0.947** $\pm$0.010	0.861 $\pm$0.014	0.850 $\pm$0.015	**0.084** $\pm$0.011	0.313 $\pm$0.014	0.269 $\pm$0.015
elephant	**33.91** $\pm$2.04	26.94 $\pm$0.45	28.34 $\pm$0.55	**0.922** $\pm$0.033	0.731 $\pm$0.012	0.772 $\pm$0.013	**0.088** $\pm$0.013	0.423 $\pm$0.006	0.325 $\pm$0.010
flamingo	**34.96** $\pm$0.65	25.74 $\pm$0.73	27.10 $\pm$1.01	**0.928** $\pm$0.007	0.753 $\pm$0.018	0.783 $\pm$0.020	**0.106** $\pm$0.015	0.483 $\pm$0.008	0.349 $\pm$0.013
hike	**29.74** $\pm$1.88	25.15 $\pm$0.25	24.77 $\pm$0.38	**0.886** $\pm$0.048	0.698 $\pm$0.019	0.682 $\pm$0.022	**0.108** $\pm$0.026	0.388 $\pm$0.019	0.343 $\pm$0.017
horsejump-high	**34.78** $\pm$1.78	28.35 $\pm$0.41	27.28 $\pm$0.64	**0.932** $\pm$0.016	0.846 $\pm$0.019	0.830 $\pm$0.020	**0.074** $\pm$0.013	0.249 $\pm$0.023	0.226 $\pm$0.024
kite-surf	**37.96** $\pm$0.50	28.04 $\pm$0.70	27.88 $\pm$0.32	**0.949** $\pm$0.005	0.780 $\pm$0.026	0.780 $\pm$0.018	**0.068** $\pm$0.006	0.420 $\pm$0.031	0.400 $\pm$0.016
kite-walk	**37.96** $\pm$0.66	29.44 $\pm$0.38	29.58 $\pm$0.56	**0.941** $\pm$0.010	0.804 $\pm$0.009	0.818 $\pm$0.011	**0.070** $\pm$0.012	0.367 $\pm$0.007	0.334 $\pm$0.014
libby	**38.89** $\pm$0.56	29.62 $\pm$0.94	29.35 $\pm$0.76	**0.949** $\pm$0.004	0.819 $\pm$0.028	0.828 $\pm$0.028	**0.095** $\pm$0.010	0.399 $\pm$0.031	0.342 $\pm$0.025
lucia	**30.90** $\pm$1.44	26.03 $\pm$0.54	26.63 $\pm$0.65	**0.869** $\pm$0.047	0.690 $\pm$0.027	0.742 $\pm$0.036	**0.178** $\pm$0.068	0.407 $\pm$0.033	0.329 $\pm$0.040
motorbike	**37.42** $\pm$0.89	27.33 $\pm$0.93	29.33 $\pm$1.10	**0.950** $\pm$0.008	0.779 $\pm$0.023	0.843 $\pm$0.014	**0.082** $\pm$0.011	0.376 $\pm$0.020	0.241 $\pm$0.017
swing	**35.70** $\pm$0.89	26.14 $\pm$0.54	27.88 $\pm$0.50	**0.926** $\pm$0.010	0.722 $\pm$0.021	0.808 $\pm$0.019	**0.119** $\pm$0.017	0.404 $\pm$0.027	0.289 $\pm$0.029
tennis	**35.65** $\pm$4.22	27.43 $\pm$1.89	28.81 $\pm$1.32	**0.928** $\pm$0.036	0.806 $\pm$0.062	0.862 $\pm$0.044	**0.120** $\pm$0.049	0.328 $\pm$0.081	0.209 $\pm$0.054
Mean	**35.35** $\pm$1.30	27.31 $\pm$0.66	28.09 $\pm$0.66	**0.929** $\pm$0.018	0.763 $\pm$0.023	0.800 $\pm$0.020	**0.098** $\pm$0.020	0.390 $\pm$0.024	0.302 $\pm$0.021

Assessment (BVQA), particularly for generative video models [37]. While earlier methods are tied to specific image and video corruptions [8, 28, 27], *driving* the need for combining multiple scores, newer feature- and learning-based approaches [2, 26, 35] are often model- or domain-dependent and have exhibited questioned robustness [1].

Secondly, assessing the quality of the decomposition or texture edits itself is highly non-trivial. For scene decomposition, removing an object requires the system to hallucinate the previously occluded geometry and appearance. Since the content of the occluded region (including potential changes) cannot be known without a geometric reference, any arbitrary, visually plausible content may be valid, which greatly complicates traditional quality scoring. Furthermore, for texture edits, separating the quality contribution of the *continuous texture application* (method influence) from the inherent quality of the *user-defined texture* (user influence) presents an additional difficulty, as the latter can heavily skew the rated video quality.

While assessing the perceptual quality of the edits is challenging, an additional evaluation of the temporal consistency of the edited video may provide insights. This approach allows us to quantify the fluctuations introduced during the editing process. To this end, we computed the Temporal

Table 6: Quantitative Comparison of Temporal Consistency (FVD and T-LPIPS) against core baselines on a selected subset of the Waymo and DAVIS datasets. Results are reported as mean metric value and standard deviation (± STD) across (sub-)sequences. Our method demonstrates favorable FVD scores and achieves T-LPIPS closest to the Ground Truth (GT), indicating both high quality and inter-frame stability. (Best results are shown in bold.)

Method	Waymo		DAVIS	
	FVD ↓	T-LPIPS	FVD ↓	T-LPIPS
Ours	**174** ± 238	0.063 ± 0.031	**108** ± 35	0.143 ± 0.022
ORe	423 ± 446	0.055 ± 0.029	N/A	N/A
ERF	439 ± 350	0.053 ± 0.026	N/A	N/A
ORF	N/A	N/A	986 ±153	0.104 ± 0.023
LNA	N/A	N/A	595 ± 38	0.116 ± 0.022
GT	N/A	0.079 ± 0.029	N/A	0.155 ± 0.020

Table 7: Temporal Consistency of Editing Figures

Edit	T-LPIPS			FID [9]			
	Ours	ORe	GT	Ours		ORe	GT	
Fig. 3, Seg. 125	0.052 ± 0,027	0.054 ± 0,032	0.081 ± 0,033	2.244 ± 1.857		1.972 ± 1.326	2.228 ± 1.578	
Fig. 3, Seg. 141	0.056 ± 0.048	0.063 ± 0.062	0.103 ± 0.072	0.999 ± 1.032		1.111 ± 1.308	1.517 ± 1.222	
Fig. 4	0.037 ± 0.006	N/A	0.043 ± 0.006	0.175	-	N/A	0.173	-
Supl. Fig. 12	0.249 ± 0.028	N/A	0.261 ± 0.026	0.255	-	N/A	0.185	-
Supl. Fig. 13	0.058 ± 0.012	N/A	0.080 ± 0.012	0.132	-	N/A	0.180	-

(T)-LPIPS score (frame-by-frame perceptual difference) and the Fréchet Inception Distance (FID) [9] score, applied frame-by-frame wise to the edited video sequence. These scores provide a quantitative measure of temporal stability for the decomposed objects or edited regions, which, combined with a qualitative human assessment, forms the basis of our edit evaluation, as shown in Tab. 7. For T-LPIPS we report the mean and standard deviation over all image-pairs and object regions in case of Fig. 3, while over the full image when editing, incorporating background edits. For FID we report the standard deviation only for the per-object decomposition evaluation, given that it is computed as a distributional measure. For all stated edits, our consistency results are comparable to our reference method or the respective ground truth (GT).

C.3 Evaluation on Large Ego Motion

Our method was originally designed with a primary focus on texture editable scenes, often implying lower ego-motion to maintain stable views of objects. However, to rigorously test the robustness of our motion model, we conducted an evaluation on two additional Waymo Open Dataset sequences, s-191 and s-254 [4], which feature significant camera ego-motion (visualized in Fig. 2).

Following our analysis on other sequences, we evaluated the performance by focusing on foreground objects managed by dedicated nodes (vehicles and humans) and by assessing overall image quality.

For foreground objects (Vehicles and Humans), which are managed by dedicated nodes with a robust rigid motion model, our approach achieves substantial improvements—up to 8 dB PSNR—compared to all baselines (see Tab. 8). This success confirms that our rigid motion model, being independent of the large global camera motion, is suited for handling moving objects in challenging, large ego-motion environments, provided the object's view-angle does not completely change.

The overall image scores (PSNR, SSIM, LPIPS) are presented in Tab. 9. While these composite scores are comparable to competitors ORe and ERF, they are not substantially higher. This trade-off is attributed to the inherent limitations of our 2.5D representation, as discussed in our limitation section. In large ego-motion scenes, our planar background assumption requires the flow network to learn large, complex flow vectors to cover the rapidly changing content. Assuring the fidelity of such long flow vectors purely through photometric loss proves highly challenging. In failure cases,

[4] Referring to segment-1918764220984209654 and segment-2547899409721197155, subdivided in 3 / 4 sub-segments.

Figure 2: Ground truth references and mask examples out of the two tested *large-ego-motion* Waymo sequences [24] (s-191, s-254).

the flow tends to collapse regions onto the plane and unfold them as needed, which unfortunately introduces characteristic *wavy-line artifacts* within the background. We showcase these artifacts along with the still highly competitive foreground objects in Fig. 3.

This evaluation demonstrates a clear and informative trade-off. We emphasize that our proposed Neural Atlas Graph Model is fundamentally 2.5D (planes + flow), endowed with a large inductive bias that excels at regularizing solutions for low-parallax or sparsely observed scene elements—a strength evident in the robust handling of moving foreground objects and in providing direct texture editability. However, the background artifacts observed within large ego-motion scenes reveal the current limitations of this 2.5D planar representation, which is not as well suited to model complex 3D structures with large amounts of self-occlusion within the background. We believe these findings strongly indicate that extending this architecture towards a hybrid object-centric graph model — combining both 2D and 3D primitives in a single render graph — is a compelling direction for future research.

Table 8: Quantitative Evaluation on Large-Ego-Motion Dynamic Driving Sequences of the Waymo [24] Open Driving Dataset on Vehicle and Human class.

Seq.	Vehicle PSNR ↑			Vehicle SSIM ↑			Human PSNR ↑			Human SSIM ↑		
	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF
s-191	**37.60**	31.20	25.78	**0.959**	0.928	0.714	**34.40**	27.34	22.93	**0.908**	0.808	0.589
s-254	**35.07**	30.52	26.00	**0.926**	0.928	0.735	**34.42**	28.87	22.67	**0.919**	0.880	0.629

Table 9: Quantitative Evaluation on Large-Ego-Motion Dynamic Driving Sequences of the Waymo [24] Open Driving Dataset.

Seq.	PSNR ↑			SSIM ↑			LPIPS ↓		
	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF
s-191	32.02	**33.17**	29.97	0.892	**0.952**	0.864	0.209	**0.088**	0.244
s-254	31.70	**31.91**	29.32	0.911	**0.950**	0.871	0.190	**0.093**	0.241

C.4 Additional Ablation Experiments

To assess the contribution of different components of our proposed model, we conducted a comprehensive ablation study on a subset (s-141, s-975) of the Waymo Open Dataset [24]. These sequences were further divided into 8 subsequences, which, given a systematic evaluation of all key model components and hyperparameters, yielded 96 additional experiments. The results of these experiments are detailed in Tab. 10. The top row of the table, labeled "Large", represents the performance of our full reference model as described in the main manuscript.

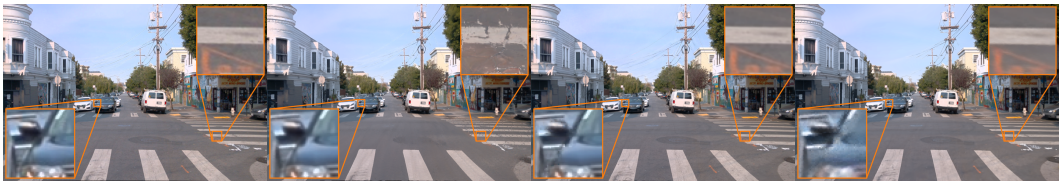

Figure 3: Visual quality comparison on the *large-ego-motion* scene s-191. A clear trade-off is observed. Foreground objects managed by the rigid motion model exhibit increased sharpness and reduced edge artifacts compared to baselines. Conversely, the high flow compensation required by our 2.5D background may cause visual degradation in rapidly changing background regions, manifesting as flow artifacts or blurring.

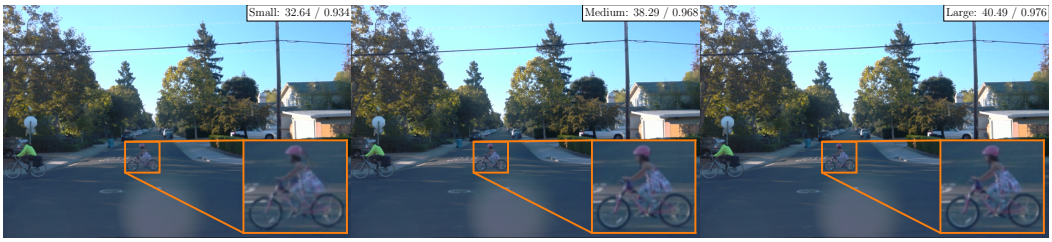

Figure 4: Representative examples of NAG nodes with varying parametrization sizes (Small, Medium, Large), including their PSNR / SSIM scores. Noticeable image quality degradation and flow collapsing artifacts are evident in the small node due to its limited representation, whereas distinguishing visual differences between medium and large nodes is challenging, with only minor lighting variations on the ground.

Parametrization Sizes We evaluated the impact of varying the model size ("Medium" and "Small")[5] and observed a general trend of performance degradation (lower PSNR and SSIM, higher LPIPS) with reduced capacity, highlighting the importance of model scale for achieving optimal reconstruction quality. A representative visual example of these different model sizes and their corresponding PSNR/SSIM scores can be found in Fig. 4, further illustrating the qualitative differences.

Initialization & Coarse-to-fine Furthermore, we investigated the significance of several key modules within our architecture by systematically excluding or modifying them. The rows "Coarse Init-Projection" and "Excl. Coarse-to-fine" examine the role of our coarse initialization and the subsequent coarse-to-fine refinement strategy. For the first, we limit the size of our initial estimates for color $\tilde{C}_i \in \mathbb{R}^{20\times20}$ and opacity $\tilde{A}_i \in \mathbb{R}^{20\times20}$ to a much lower spatial extend than the original used, which is based on the mask size. This shall mimic a mean initialization of the objects. The latter, deactivates our coarse-to-fine scheme. While the performance drop observed may look rather small, the visual changes on decomposition and edits may be very significant, as excluding these components could lead to much more background information in the foreground or vice-versa.

Flow- & View-Fields The rows "Excl. Flow" and "Excl. View-Dependence" quantify the impact of our optical flow estimation and view-dependent modeling components, by disabling them respectively. The substantial decrease in all evaluated metrics upon their removal underscores their critical role in handling motion and viewpoint changes within the driving scenes. Notably, the combined exclusion of both flow and view-dependence ("Excl. Flow + View-Dependence") resulted in the most significant performance decline, emphasizing the synergy between these modules.

Position Learning We also assessed the translation learning component ("Excl. Translation Learning"). Excluding this learning component slightly weakens the overall reconstruction quality.

[5]The sizes "Large", "Medium", "Small" are referring to different parameterizations of our MLP Network and Hash-Grid Configurations. Effectively, they are reducing the number of levels and sizes within the hash-grid encoding, as well as reducing the number of hidden layers within our MLPs. For details we refer to our code base.

Table 10: Ablation Experiments. We conducted ablation studies on a subset of our Waymo Datasets [24], evaluating various components of our model. Best results are **bold**, second best are underlined. The top row (Large) marks the reference model stated in our manuscript. On different model sizes, the scores may degrade significantly. When excluding or changing certain keyparts, we observe degradation of the performance, showing their importance. When also learning the plane rotation (cf. Davis), this slightly benefits performance reported on this subset.

Abl.	PSNR ↑	SSIM ↑	LPIPS↓	Vehicle		Human	
				PSNR↑	SSIM↑	PSNR↑	SSIM↑
Large	41.42	**0.977**	0.057	44.94	0.986	44.65	0.987
Medium	39.33	0.968	0.071	42.09	0.973	41.80	0.975
Small	35.64	0.943	0.099	36.56	0.936	36.90	0.939
Coarse Init-Projection	41.27	**0.977**	0.060	44.87	0.985	44.84	0.987
Excl. Coarse-to-fine	41.37	**0.977**	0.058	44.93	0.985	44.86	0.987
Excl. Flow	39.44	0.974	0.063	44.47	0.981	44.26	0.985
Excl. View-Dependence	38.08	0.961	0.095	34.07	0.901	34.91	0.913
Excl. Flow & View-Dependence	32.29	0.936	0.110	24.71	0.775	27.92	0.808
Excl. Translation Learning	39.37	0.971	0.062	45.15	0.984	44.88	0.987
Incl. Plane Rotation Learning	**41.46**	**0.977**	**0.056**	**45.35**	**0.987**	**46.94**	**0.992**
Num. Position CP $P = F/2$	40.44	0.972	0.061	45.22	0.986	44.77	0.986
Num. Position CP $P = F3/4$	40.97	0.976	0.060	45.08	0.986	44.66	0.986
Excl. Mask-Loss	41.31	**0.977**	0.058	44.95	0.986	44.88	0.987
Morph. Masks	41.24	0.975	0.060	44.92	0.984	44.91	0.986
Morph. Masks, Excl. Mask-Loss	41.29	0.975	0.060	45.00	0.985	44.94	0.986
Bounding. Masks	41.15	0.974	0.061	44.82	0.982	44.87	0.986
Bounding. Masks, Excl. Mask-Loss	41.20	0.975	0.060	44.80	0.982	44.88	0.986

While the quantitative impact is minor on the studied Waymo sequence, the degradation would likely be more severe if less precise initializations (e.g., non-3D bounding box initializations) were provided. The model's ability to maintain high object-based scores, despite excluding translation, suggests that the planar flow-, or view-dependent- fields compromises for errors within the rigid motion model.

Additionally, we explored the effect of explicitly learning plane rotations, similar to our DAVIS [19] experiments. The row "Incl. Plane Rotation Learning" shows a slight improvement across all metrics on this specific Waymo subset compared to the "Large" baseline. However, we were unable to consistently verify this improvement across additional Waymo sequences, suggesting that its benefit might be scene-specific or less pronounced in more diverse scenarios.

Position Granularity The two rows ("Num. Position CP $P = F/2$" and "Num. Position CP $P = F3/4$") investigate the influence of the number of control points used for our motion model. Employing fewer control points results in a smoother motion trajectory for both the ego-camera and individual objects. Interestingly, the observed improvement in Vehicle PSNR and SSIM with fewer control points is relatively minor and just slightly holds for the Human category, which often exhibits more complex, non-rigid motion. This discrepancy suggests that while a smoother motion constraint might offer a slight benefit for predominantly rigid objects like vehicles, it could be insufficient and potentially detrimental for capturing the intricate deformations and trajectories of non-rigid objects such as pedestrians. Further, over-smoothing the camera motion does negatively impact overall scene alignment, outweighing any minor per-object benefits seen for vehicles, measured in the worse overall scores.
Yet, the observed benefits suggest that imposing different smoothness assumptions for rigid objects (like vehicles), non-rigid objects (like pedestrians), as well as the ego-camera, may further improve the overall reconstruction quality, but requires further investigation.

Mask Quality To assess the influence of initial mask quality and the mask loss term on our reconstruction, we conducted an exemplary ablation. First, we tested the impact of the mask loss itself by setting its weight to 0 on the original data, labeled "Excl. Mask-Loss" in Tab. 10.

Second, we generated corrupted mask versions from our precise segmentations to simulate less ideal input and training conditions. We created two specific mask types for this analysis: the *Morphological Masks* ("Morph. Masks") simulate imprecise segmentation by artificially corrupting the precise masks using morphological operations—specifically, smooth boundary erosion and dilation based on Perlin noise [20] maps. Alternatively, the *Bounding Box Masks* ("Bounding. Masks") simulate the scenario where only axis-aligned bounding boxes are available, by using the bounding box of the original object mask. These two corrupted mask versions were then each trained both with and without the mask loss, resulting in the final set of conditions detailed in Tab. 10. Examples of these corrupted masks are presented in Fig. 5.

Based on the quantitative metrics in Tab. 10, we observe that the mask quality and the mask loss term do not significantly affect the overall reconstruction quality (PSNR / SSIM / LPIPS). This relative resilience is expected, as the highly over-parametrized, node-based representation of the NAG has sufficient capacity to fit the scene even with minor initialization errors. However, we explicitly introduce the mask loss to suppress noise in the foreground and increase opacity in low-contrast segmentation areas (e.g., a grey car on a grey road).

Since the masks are used for initializing and refining individual atlas nodes to correctly factorize the scene, performance differences become apparent when qualitatively studying the decomposed objects. We present two decomposed objects from the s-141 scene in Fig. 6 and 7. For highly contrastive objects with clear, independent motion (like the white truck in Fig. 6), the decomposition quality is minimally impacted by mask quality or loss. At most, a slight increase in opacity around the object boundaries fitting to background noise can be observed. Conversely, for challenging, occluded objects (such as the person in Fig. 7), the effect is significant: the representation severely degrades when using aberrated masks. The original precise masks were necessary to maintain a reasonable representation of the person, while disabling the mask loss further exacerbated the fitting to noise.

Furthermore, poor mask quality may indirectly affect texture editability, as fitting exterior noise or background content into the atlas can push information into the view-dependence field, potentially hindering subsequent texture modification. Nevertheless, the qualitative examples in Fig. 6 suggest that even using bounding box masks or segmentation models with coarser outputs could be sufficient to yield a reasonable scene decomposition when interest lies primarily in dominant foreground objects with clear motion patterns. While an in-depth discussion on our method's mask-quality sensitivity would require dedicated experiments on synthetic data, our exemplary study indicates certain usability even in the absence of precise segmentations, relying only on bounding boxes.

Figure 5: Ground truth references and masks (top) and their corrupted versions using morphological operations (middle) as well as axis-aligned bounding box masks (bottom). We showcase four frames of the s-141 sequence (timestamps 40, 45, 50, 55), with 0.5 seconds spacing. The morphological masks show significant aberated and time-varying borders, while the imprecision of bounding boxes pose a challenge on adressing overlapping.

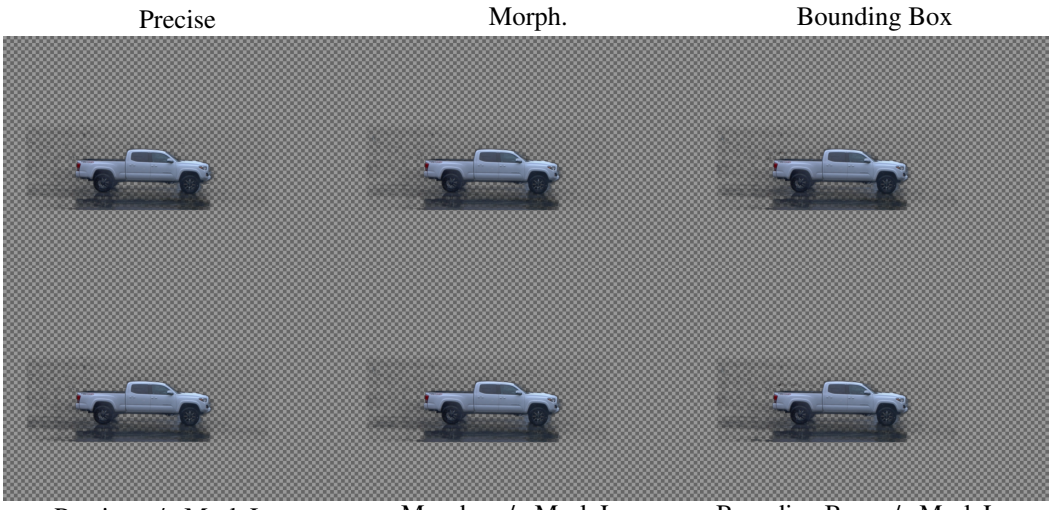

Figure 6: Object decomposition of the white truck from scene s-141 (timestamp 45; ref. Fig. 5). The top row shows results trained with the mask loss, while the bottom row excludes it. Even under aberrated masks (Morph. and Bounding Box), the decomposition remains highly precise for contrast-rich and independently moving objects, showing only minor increases in background noise fitting.

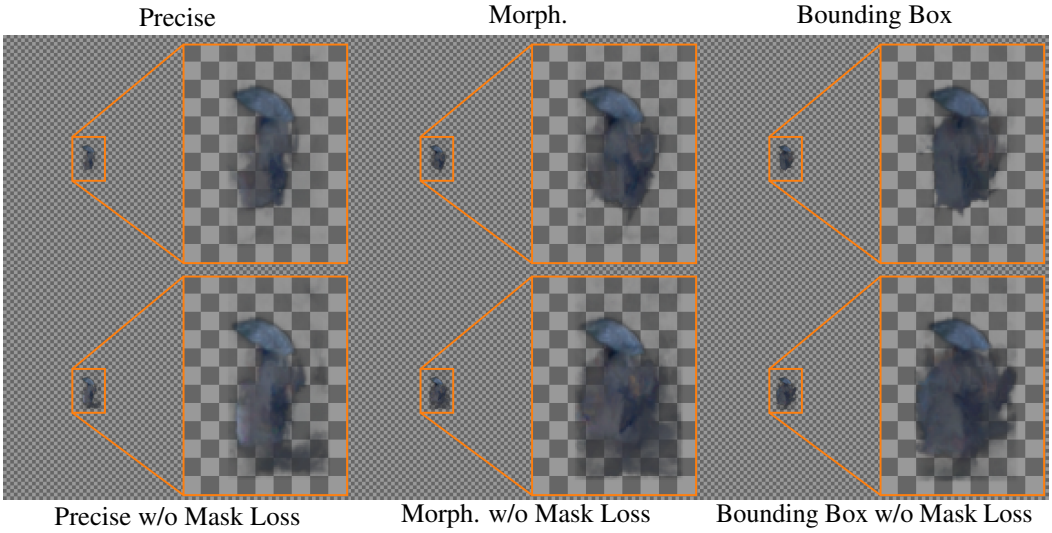

Figure 7: Object decomposition of an occluded person in scene s-141 (following Fig. 6). While the original version maintains a consistent silhouette despite heavy occlusion, the use of aberrated masks (Morph. and Bounding Box) tends to produce a less consistent and visually disturbed representation in these challenging occluded regions.

In summary, our ablation studies provide valuable insights into the contribution of individual components of our model, highlighting the importance of model size, flow estimation, view-dependent modeling, and translation learning for achieving high-quality reconstructions.

C.5 Additional Visual Results

This supplementary section provides extended visual results that further illustrate the capabilities of our proposed NAG representation. We present additional examples showcasing the editing potential of NAGs, including object insertion, retiming, and shifting. Furthermore, we offer supplementary visual examples from the Davis Dataset [19], including reconstructions and their corresponding scene decompositions, providing deeper insights into our model's performance and representation.

In Fig. 8 we demonstrate our reconstructions on 3 more scenes, showcasing its handling of complex and fine details like water droplets, the feets of cyclists which we are, despite their challenging motion, capable to represent accurately. Further we highlighted our improved handling of distant objects emphasizing our visual performance increases w.r.t our baselines.

Further, we illustrate in Fig. 9 the comprehensive editing capabilities of our method on the Waymo s-125 scene. We demonstrate three distinct types of manipulations: object removal (specifically, the truck on the left), object duplication / adding and precise spatial shifting (exemplified by the white car copied and moved by 2 units to the left and 0.5 units towards the camera), and temporal manipulation (achieved by duplicating the red car and shifting its presence by ±5 timestamps). These examples collectively highlight our method's versatility in handling edits that remain consistent with our flow- and view-dependent model, allowing for precise control over scene composition and dynamics.

Figure 10 showcases the visual effectiveness of our method on the DAVIS Dataset [19]. Our view-dependent model components lead to significant improvements in visual quality compared to baselines, evident in increased sharpness and the detailed rendering of fine structures such as tire spokes. Notably, even with its view-dependent nature, our method produces reasonable background estimates in occluded regions, as illustrated by the car-shadow example. Figure 11 presents results on three additional challenging DAVIS sequences where our method outperforms baselines: 1) motorbike: capturing the fast-moving foreground with fidelity; 2) bear: accurately modeling non-rigid motion and intricate fur texture; and 3) hike: handling actor movement against a complex, high-depth background.

Lastly, we state two more texture edits of DAVIS [19] sequences in Fig. 12 and Fig. 13 where we utilized an off-the-shelf image generation model to create new textures for the decomposed foreground object, and applied these consistently along the video, yielding accurate edits even when changing the complete texture of these mostly rigid moving objects.

C.6 Additional Gaussian Splatting Baselines

While ORe and ERF are recent 3DGS and NeRF baselines for object-specific and agnostic scene reconstruction, we broaden our analysis to include further dynamic 3D Gaussian Splatting methods.

To ensure comprehensive coverage against the state-of-the-art methods in dynamic 3DGS, we specifically evaluated three additional methods: Street Gaussians Street Gaussians (SGS) [30] (object-specific), Periodic Vibration Gaussian (PVG) [3] (object-agnostic) and Deformable 3D Gaussians (DGS) [32], which uses a global, non-separable gaussian representation, on our selected Waymo sequences. We utilize the benchmark suite from [4] for evaluation, with overall results presented in Tab. 11. Our method, NAG, outperforms all these additional baselines in terms of PSNR (+4.08 dB) and SSIM (+0.014), while remaining competitive in LPIPS (+0.005). Furthermore, the object-specific results in Tab. 12 confirm NAG's superior performance in preserving structural details and enhancing dynamic object representation.

We note, that PVG yields a 0.85 dB higher PSNR score than ORe. Yet, given PVG is an object-agnostic representation, while ORe is object-specific, with support for object-based editing, more closely matching our methods capabilities, we stick to ORe as our main GS comparison within our manuscript.

C.7 Training Time

The training duration for the Neural Atlas Graph ranges from 2 to 6 hours per scene on an NVIDIA L40 GPU. To accurately assess this performance, we contextualize these training times within the domain of dynamic neural scene representation and we further detail the sources of this computational cost. State-of-the-art methods, such as ORe [4], report comparable training durations (e.g.,

Figure 8: Extended visual results on the Waymo Dataset [24], showcasing reconstructed sequences s-141, s-975, and s-125. Our model demonstrates the ability to capture fine details, such as water droplets and the non-rigid motion of cyclists' feet, while exhibiting fewer artifacts compared to baseline methods.

Figure 9: Illustration of editing operations on the s-125 scene. We showcase: the removal of an existing object (left truck), copying and spatially shifting the white car, and the temporal manipulation of the red car through duplication and a ±5 timestamp shift.

approximately one hour for a single scene at a lower 960×640 resolution). Crucially, the proposed method consistently trains at full resolution (1920×1280 for Waymo and up to 1920×1080 for DAVIS), which substantially increases the computational load compared to baselines often optimized for lower resolutions. Furthermore, highly optimized methods, such as those leveraging Gaussian Splatting, have benefited from dedicated native CUDA implementations and extensive optimization efforts over recent years [11, 34].

Tab. 13 provides a direct comparison of observed training times across all baselines at their full respective resolutions. For Waymo, we used a subsequence from segment 975 with a length of 68 frames, and for DAVIS, the popular *bear* sequence with 74 frames.

Our training time is significantly shorter than the other atlas-based methods, LNA and ORF, on the DAVIS dataset. On Waymo, our time is comparable to ORe [4] but slower, which is attributable to the latter's highly optimized Gaussian Splatting implementations. ERF [31] is significantly faster as it is not a scene graph method and lacks a dedicated model per object, reducing architectural overhead. It should also be noted that ERF generally produced less accurate results in our quantitative experiments.

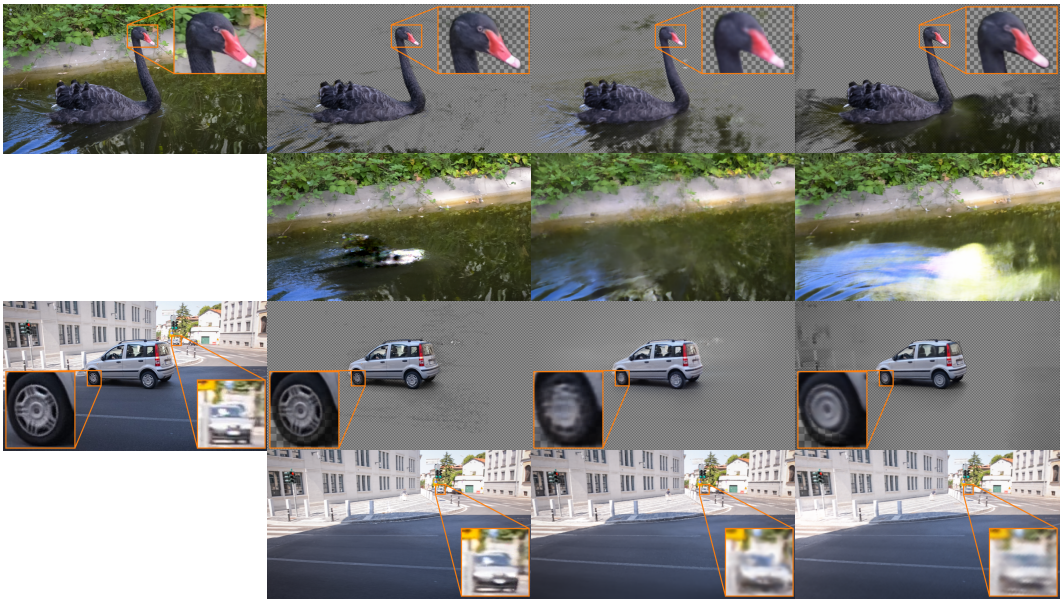

Figure 10: Visual comparison and decomposition of the blackswan and car-shadow sequence within the DAVIS dataset. The insets stating difficult regions underlining the capabilities of our model in accurately representing highly textured regions (swan head), time-variant content (spinning wheels) and distant background objects.

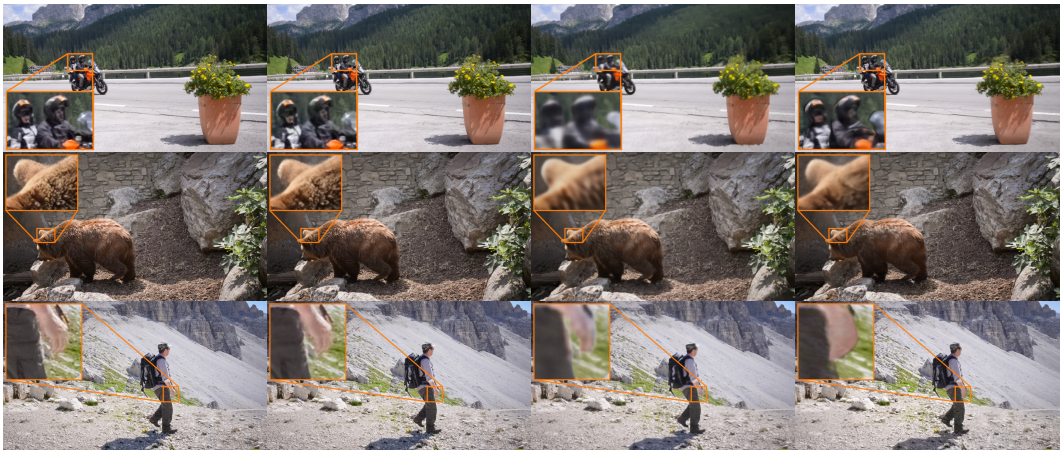

Figure 11: Additional visual examples of the DAVIS [19] sequences motorbike, bear and hike, showcasing our models quality in representing fine and complex details on rigid and non-rigid foreground actors.

We emphasize that three key components are essential for achieving the high-quality and editable NAG representation, inherently contribute to the overall training duration:

1. **Extensive Ray-Casting**: Learning dynamic video at high resolution necessitates extensive ray-casting. For instance, our full training involves 2.8×10^9 rays over 80 epochs. Training time naturally reduces for lower resolutions or shorter videos (e.g., to under 20 minutes for highly reduced settings).
2. **Multi-Stage Optimization**: A three-phase optimization strategy is employed to ensure stable convergence, accurate decomposition, and enhanced texture editability, yet may add an overhead in execution time.

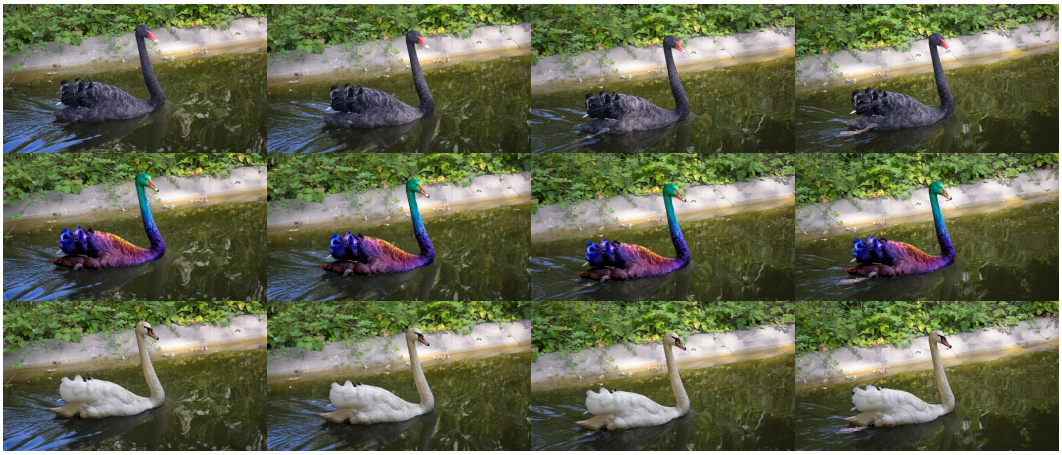

Figure 12: Advanced texture editing applied to the blackswan sequence (DAVIS dataset [19]). The top row presents the ground truth. The lower rows display edits where an off-the-shelf image generation model was leveraged to create rainbow and white swan texture variants. Using these generated textures, our method effectively propagates these localized changes consistently across all video frames, demonstrating robust temporal coherence.

Figure 13: Texture edits using DAVIS [19] boat sequence. Similar to Fig. 12, we state the boat sequence, retexturing it with a rainbow and a red texture. The top row shows the ground truth, while the lower ones are the respective edits.

3. **Per-Object Networks**: The training time scales with scene complexity due to the presence of dedicated, independent neural networks for each object and the background, increasing the total parameter count and computation per step.

These architectural choices are a necessary investment to deliver the decomposition and editing capabilities that are the focus of this work. Full training details and ablation studies are available in the Experimental Details (Sec. B.2) of this supplementary material. We see potential areas for future optimization in all these components, such as leveraging more efficient ray-casting implementations, employing object size-driven network architectures, or improving initialization strategies to significantly reduce training time.

D Discussion on Scene Representation Methods

As recent literature has introduced a wide range dynamic scene representation methods tailored to different domains, in this section we summarize their core ideas and priors to situate Neural Atlas Graphs (NAGs) within the broader research context. We separate these broadly into 2D-, 2.5D, and

Table 11: Quantitative Evaluation on Dynamic Driving Sequences of the Waymo [24] Open Driving Dataset. The temporal consistency is measured by the inter-frame standard deviation ($\pm$ STD), which is calculated over sub-segments and mean-aggregated per sequence. Best results are in bold. PVG refers to Periodic Vibration Gaussian [3], DGS to Deformable 3D Gaussians [33], and SGS to Street Gaussians[30]. Our method compares favorably against these additional object-agnostic (PVG, DGS) and object-specific (SGS) baselines, including high consistency in PSNR, SSIM and LPIPS.

Seq.	PSNR ↑				SSIM ↑				LPIPS ↓			
	Ours	PVG	DGS	SGS	Ours	PVG	DGS	SGS	Ours	PVG	DGS	SGS
s-975	**40.21** ±1.11	34.00 ±1.88	36.67 ±1.96	36.19 ±2.24	**0.976** ±0.004	0.958 ±0.006	0.961 ±0.007	0.962 ±0.005	0.058 ±0.012	0.064 ±0.008	0.059 ±0.018	**0.054** ±0.010
s-203	**43.15** ±0.39	38.71 ±0.98	35.86 ±1.03	33.08 ±1.81	**0.978** ±0.001	0.966 ±0.001	0.960 ±0.002	0.958 ±0.003	0.070 ±0.004	**0.052** ±0.002	0.063 ±0.004	0.060 ±0.004
s-125	**43.32** ±0.49	38.78 ±0.61	35.82 ±1.16	38.47 ±0.72	**0.980** ±0.003	0.964 ±0.001	0.958 ±0.004	0.964 ±0.002	0.057 ±0.007	0.046 ±0.002	0.053 ±0.005	**0.040** ±0.002
s-141	**42.55** ±1.60	38.31 ±0.54	33.97 ±1.06	33.56 ±1.21	**0.978** ±0.003	0.963 ±0.002	0.949 ±0.005	0.954 ±0.003	0.057 ±0.005	**0.054** ±0.004	0.080 ±0.010	0.065 ±0.005
s-952	**41.89** ±0.59	39.55 ±0.79	35.62 ±1.59	34.78 ±1.38	**0.976** ±0.003	0.968 ±0.002	0.963 ±0.004	0.961 ±0.003	0.058 ±0.006	**0.041** ±0.003	0.047 ±0.008	0.047 ±0.004
s-324	**40.85** ±1.31	36.97 ±0.65	34.01 ±1.25	30.96 ±2.10	**0.977** ±0.002	0.961 ±0.002	0.953 ±0.004	0.942 ±0.010	**0.038** ±0.004	**0.038** ±0.003	0.049 ±0.006	0.059 ±0.009
s-344	**41.84** ±0.52	38.47 ±0.72	32.75 ±1.32	29.90 ±1.01	**0.983** ±0.001	0.968 ±0.002	0.957 ±0.004	0.953 ±0.004	0.031 ±0.002	**0.030** ±0.002	0.046 ±0.005	0.050 ±0.006
Mean	**41.85** ±0.91	37.77 ±0.87	34.83 ±1.36	33.70 ±1.49	**0.978** ±0.002	0.964 ±0.003	0.957 ±0.004	0.956 ±0.005	0.051 ±0.006	**0.046** ±0.003	0.056 ±0.008	0.054 ±0.006

Table 12: Quantitative Evaluation of Human and Vehicle Rendering on Waymo [24] Driving Sequences for additional 3DGS methods. The temporal consistency is computed as in Tab. 11 - by the inter-frame standard deviation ($\pm$ STD), calculated over sub-segments and mean-aggregated per sequence.

Seq.	Vehicle PSNR ↑				Vehicle SSIM ↑				Human PSNR ↑				Human SSIM ↑			
	Ours	PVG	DGS	SGS	Ours	PVG	DGS	SGS	Ours	PVG	DGS	SGS	Ours	PVG	DGS	SGS
s-975	**46.79** ±1.21	33.11 ±1.82	27.22 ±2.55	32.75 ±2.97	**0.991** ±0.001	0.912 ±0.025	0.824 ±0.064	0.930 ±0.031	**45.37** ±1.58	33.06 ±1.90	26.09 ±2.64	23.01 ±2.93	**0.989** ±0.002	0.903 ±0.026	0.788 ±0.049	0.744 ±0.048
s-203	**41.90** ±1.89	34.46 ±2.05	28.39 ±2.94	30.19 ±2.96	**0.986** ±0.005	0.946 ±0.019	0.848 ±0.063	0.897 ±0.050	**45.40** ±1.65	35.79 ±1.50	27.28 ±2.88	16.22 ±2.57	**0.986** ±0.004	0.947 ±0.006	0.870 ±0.031	0.665 ±0.067
s-125	**41.00** ±1.90	33.61 ±1.28	25.97 ±1.95	28.82 ±2.39	**0.989** ±0.004	0.955 ±0.011	0.815 ±0.053	0.875 ±0.054	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
s-141	**43.21** ±1.44	33.76 ±1.84	26.15 ±2.29	32.07 ±2.34	**0.981** ±0.007	0.936 ±0.015	0.801 ±0.050	0.916 ±0.032	**44.22** ±1.61	36.13 ±1.67	29.78 ±3.31	25.08 ±2.91	**0.986** ±0.005	0.940 ±0.016	0.828 ±0.070	0.725 ±0.098
s-952	**40.94** ±1.46	32.88 ±1.97	27.27 ±3.70	28.74 ±3.88	**0.986** ±0.004	0.942 ±0.016	0.848 ±0.063	0.894 ±0.055	**40.45** ±2.82	34.90 ±1.77	26.02 ±3.29	23.29 ±2.77	**0.968** ±0.021	0.928 ±0.018	0.706 ±0.087	0.629 ±0.092
s-324	**41.71** ±1.56	34.03 ±2.19	29.20 ±3.29	29.29 ±3.73	**0.986** ±0.004	0.948 ±0.016	0.869 ±0.057	0.880 ±0.062	**44.12** ±1.95	34.16 ±2.21	26.20 ±2.69	22.56 ±2.40	**0.988** ±0.005	0.918 ±0.027	0.727 ±0.076	0.626 ±0.098
s-344	**43.97** ±1.69	34.59 ±1.66	28.59 ±2.77	28.81 ±3.35	**0.985** ±0.007	0.948 ±0.010	0.823 ±0.031	0.860 ±0.032	**40.99** ±2.97	33.99 ±1.60	21.37 ±2.60	17.04 ±2.35	**0.975** ±0.016	0.933 ±0.014	0.640 ±0.086	0.509 ±0.095
Mean	**42.88** ±1.56	33.74 ±1.83	27.54 ±2.81	30.10 ±3.13	**0.986** ±0.005	0.940 ±0.016	0.832 ±0.054	0.893 ±0.045	**42.94** ±2.21	34.63 ±1.81	25.93 ±2.92	21.82 ±2.64	**0.981** ±0.010	0.927 ±0.019	0.740 ±0.074	0.639 ±0.088

3D model categories based on their representation domain. Furthermore, we classify them based on whether they are object-specific (separating objects based on modality/user input) or object-agnostic (differentiating only static and dynamic scenbe content). These classifications only serve as broad guidelines as their boundaries can be fluid.

2D - Scene Models This family of models focuses primarily on 2D decomposition, rooted in classic video layer separation and sprite-based techniques. The key idea here is to decompose an abritrary input video into a (static) background and one or more dynamic layers.

Layered Neural Atlases (LNA) [10] achieve this through a time-consistent *atlas* or *canvas* representation. LNA learns a 2D-to-2D warp that maps scene points onto this unwrapped atlas. This design results in a highly compact video representation that facilitates direct appearance editing on the 2D atlas – similar to editing a regular image – and propagating those changes consistently across all time

Table 13: Observed training times at full resolution for a representative sequence within the datasets.

Dataset	Method	Time (min.)
Waymo	Ours	140
	ORe [4]	127
	ERF [31]	42
Davis	Ours	198
	ORF [13]	279
	LNA [10]	444

steps. LNA is highly self-contained, requiring only the input video, masks for designated foreground layers, and pre-computed optical flow. Given that masks are used to separate objects, LNA can be treated as object-specific, relying on a separate masking pipeline. Subsequent works like Deformable Sprites [33] take a similar approach, but learn a sprite-based deformation and rely on motion cues rather than explicit masks to separate individual layers.

2.5D - Mixed Scene Models This category is comprised of models which integrate elements of 3D reasoning, such as shared reference frames or volumetric models, yet remain grounded in a predominantly layered or planar framework.

As an example, Omnimatte [16] focuses on separating objects along with their associated effects (shadows, dust, reflections). Omnimatte uses inputs similar to LNA but computes frame-by-frame homographies to yield a common reference system, which places it in the 2.5D domain due to its reliance on 3D planar motion assumptions while not explicitly making use of a 3D mesh representation. The layers are typically represented as U-Nets [21], allowing for high fidelity, but they lack the direct, intuitive editing control of LNA's atlas structure.

OmnimatteRF (ORF) [13], a direct successor of Omnimatte, addresses the limited expressivity of a 2D background layer by replacing it with a volumetric 3D NeRF. This requires the model to have a camera pose initialization, which can be derived from Structure-from-Motion (SfM) techniques such as COLMAP [22, 23] or NeRF-based approaches [14]. Similar to Omnimatte, ORF relies on optical flow, masks, homographies, and a (monucular) depth estimation – remaining object-specific.

A downside shared across these 2D/2.5D models is their reliance on a correct – often pre-defined – ordering of the foreground layers. As they compose image layers via alpha matting, an improper order can lead to transparency issues or incorrect information being picked up from objects in the background – e.g., holes in the foreground content. This becomes a large problem under varying occlusions or in busy scenes where a consistent layer ordering is hard to define as objects move in front and behind one another.

3D - Scene Models This family of models explicitly represents geometry and color content in 3D space, offering a more expressive and potentially more geometrically consistent representation of real-world scenes than 2D or 2.5D methods.

Neural Scene Graphs (NSG) [18] addresses positional editability by explicitly factorizing the scene into discrete semantic components (e.g., background, individual vehicles, pedestrians), with each object represented as a node in the scene graph. These nodes are modeled as individual radiance fields, linked by edges representing spatial relationships and 3D pose transformations. Following on NSGs, StreetGS [30] and further OmniRe (ORe) [4] use dynamic 3D Gaussian splatting techniques to represent objects, yielding faster render performance while maintaining positional editing capabilities. However, editing texture or appearance in these volumetric, neural field, or Gaussian representations is non-trivial, as changes must be propagated in a visually consistent way throughout the continuous 3D content.

The core strength of these object-specific scene graph methods is their high degree of 3D editability; objects can be translated, rotated, and removed via simple graph manipulation. Conversely, these methods are rarely self-contained, requiring extensive semantic and motion priors derived from external structured inputs such as 3D bounding boxes, object trajectories, or LIDAR depth data. Their

performance is thus heavily tied to the availability and quality of these priors, commonly found in specialized autonomous driving datasets [24, 7].

Beyond these object-specific approaches, object-agnostic methods have also emerged. These approaches, such as EmerNeRF (ERF) [31] and Gaussian Splatting approaches like Periodic Vibration Gaussian (PVG) [3], separate only static from dynamic content. While these methods have a lower reliance on data priors, they are also more difficult to directly edit as their representations can lack semantic meaning without nodes tied to discrete objects.

Neural Atlas Graphs We position our Neural Atlas Graphs (NAGs) within the 2.5D domain, drawing inspiration from 3D-based NSGs for scene decomposition while relying on lower dimensional component representations for easier appearance editing. These design choice highlight a core trade-off of all of these methods: 3D expressivity is versus ease of texture editing.

This hybrid structure enables three key capabilities: *Direct Appearance Editing* through manipulation of the 2D atlas canvas (similar to LNA); *Positional Editing* by modifying the learned spline-based rigid motion trajectories for object planes (similar to NSGs); and *Implicit Occlusion Handling* by explicitly modeling object movement and interaction along learned 3D trajectories, directly addressing the layer ordering issue.

In summary, NAGs occupy a unique point in the scene representation spectrum. NAGs adopt 3D structural priors to gain positional editing and robust occlusion handling, while retaining the ease of texture editing of a 2D atlas representations. While this approach requires sacrificing self-containment for structured priors (bounding boxes, trajectories, etc.), we showcase within our Outdoor Videos that even in domains of high self-containment (less geometric prior), NAGs construct a reasonable representation with high fidelity and demonstrate a clear advantage over 3D representations in low-ego motion and low-parallax scenes.

References

[1] Anastasia Antsiferova, Khaled Abud, Aleksandr Gushchin, Ekaterina Shumitskaya, Sergey Lavrushkin, and Dmitriy Vatolin. Comparing the robustness of modern no-reference image-and video-quality metrics to adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 700–708, 2024.

[2] Baoliang Chen, Lingyu Zhu, Chenqi Kong, Hanwei Zhu, Shiqi Wang, and Zhu Li. No-reference image quality assessment by hallucinating pristine features. *IEEE Transactions on Image Processing*, 31:6139–6151, 2022.

[3] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv:2311.18561*, 2023.

[4] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025.

[5] Ilya Chugunov, David Shustin, Ruyu Yan, Chenyang Lei, and Felix Heide. Neural Spline Fields for Burst Image Fusion and Layer Separation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25763–25773, 2024.

[6] Carl De Boor, Klaus Höllig, and Malcolm Sabin. High Accuracy Geometric Hermite Interpolation. *Computer Aided Geometric Design*, 4(4):269–278, 1987. Publisher: Elsevier.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] Rania Hassen, Zhou Wang, and Magdy MA Salama. Image sharpness assessment based on local phase coherence. *IEEE Transactions on Image Processing*, 22(7):2798–2810, 2013.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[10] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Geng Lin, Chen Gao, Jia-Bin Huang, Changil Kim, Yipeng Wang, Matthias Zwicker, and Ayush Saraf. Omnimatterf: Robust omnimatte with 3d background modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23471–23480, 2023.

[14] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023.

[16] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T. Freeman, and Michael Rubinstein. Omnimatte: Associating objects and their effects in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4507–4515, 2021.

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022.

[18] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021.

[19] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[20] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[22] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[25] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[26] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2555–2563, 2023.

[27] Zhou Wang, Alan C Bovik, and Brian L Evan. Blind measurement of blocking artifacts in images. In *Proceedings 2000 international conference on image processing (Cat. No. 00CH37101)*, pages 981–984. Ieee, 2000.

[28] Zhou Wang, Hamid R Sheikh, and Alan C Bovik. No-reference perceptual quality assessment of jpeg compressed images. In *Proceedings. International conference on image processing*, pages I–I. IEEE, 2002.

[29] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[30] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024.

[31] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. EmerneRF: Emergent spatial-temporal scene decomposition via self-supervision. In *The Twelfth International Conference on Learning Representations*, 2024.

[32] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20331–20341, 2024.

[33] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2657–2666, 2022.

[34] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

[35] Kun Yuan, Hongbo Liu, Mading Li, Muyi Sun, Ming Sun, Jiachao Gong, Jinhua Hao, Chao Zhou, and Yansong Tang. Ptm-vqa: efficient video quality assessment leveraging diverse pretrained models from the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2835–2845, 2024.

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[37] Qi Zheng, Yibo Fan, Leilei Huang, Tianyu Zhu, Jiaming Liu, Zhijian Hao, Shuo Xing, Chia-Ju Chen, Xiongkuo Min, Alan C Bovik, et al. Video quality assessment: A comprehensive survey. *arXiv preprint arXiv:2412.04508*, 2024.