Supplementary Document: 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] **Felix Heide**[2]

[1]University of Siegen [2]Princeton University

This 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 [12] and Davis [11], and also decompositions for the latter one. We choose the abbreviation ORE for OmniRe [1], ERF for EmerNeRF [14], LNA for Layered Neural Atlases [4], ORF for OmnimatteRF [6] and GT for the ground truth videos.

1 Additional Method Details

1.1 Camera Model

For casting rays into our scene, one requires a mapping from image to world coordinates. Relying on the pinhole camera model is a straightforward choice. 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:

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

Preprint. Under review.

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 of the intrinsics matrix are given by the camera's manufacturer, whereby $f \in \mathbb{R}$ defines the focal length, m_x, m_y the image width and height, and p_x, p_y is the principal point. Despite that autonomous driving datasets, the camera extrinsic are mostly given, they may include inaccuracies due to miscalibration. Further, when estimated using methods like RodynRF [7] as we do for our outdoor data experiments, they might be noisy as-well. To refine these, we utilize the same spline-based offset learning approach [2] 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 the definition:

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

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

whereby $S : [0,1] \times \mathbb{R}^P \to \mathbb{R}^F$ denotes the cubic hermite spline interpolation [3], as discussed in [2], and $\mathcal{P}^{\text{T}}_{\text{cam}}, \mathcal{P}^{\text{R}}_{\text{cam}} \in \mathbb{R}^{P\times3}$ 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 $\tilde{T}_t \in \mathbb{R}^{F\times3}$ 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 $\eta_{\text{T}} = \eta_{\text{R}} = 0.5$.

1.2 Coarse-to-fine Optimization

To limit the expressiveness of the view-dependent 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 encoding using a sparsity function [2] 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 beeing 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 $\mathcal{N}_{i,\phi} : \mathbb{R}^E \to \mathbb{R}^4$ beeing the view-dependent MLP and $\mathcal{N}_{i,f} : \mathbb{R}^E \to \mathbb{R}^{P_f\times2}$ beeing 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.

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 are sharing parameters.

2 Experimental Details

2.1 Dataset Details

Driving Scenes This section provides a detailed description of the specific subset of the Waymo Open Dataset [12] 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. 1. For the remaining images within these subsequences, we segmented all objects 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.

Table 1: Waymo [12] 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 [11], a common benchmark in video object segmentation and matting. Specifically, we selected the same 15 sequences also used by the baseline methods, ORF [6] and LNA [4], ensuring a direct basis for comparison across varied objects, backgrounds, and camera motion. We employed the provided instance masks, combining them into a single foreground mask per frame due to LNA's single-layer constraint. Consistent with ORF, we used RodynRF [7] 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 downsampled "lucia" to 960 x 540 for LNA only. For ORF, given its original compute limitations and our focus on high-resolution performance, we downsampled 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.

2.2 Training Setup

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 [1]. The training is subdivided into 3 phases. In the first phase, from epoch 0 to 5, only the positional parameters $\mathcal{P}_i^{\mathrm{T}}, \mathcal{P}_i^{\mathrm{R}}, \mathcal{P}_{\mathrm{cam}}^{\mathrm{T}}, \mathcal{P}_{\mathrm{cam}}^{\mathrm{R}}$ [2] 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 $\mathcal{P}_i^{\mathrm{T}}, \mathcal{P}_i^{\mathrm{R}}, \mathcal{P}_{\mathrm{cam}}^{\mathrm{T}}, \mathcal{P}_{\mathrm{cam}}^{\mathrm{R}}, \mathcal{F}_{i,c}, \mathcal{F}_{i,\alpha}, \mathcal{F}_{i,f}, \mathcal{F}_{i,\phi}$ are optimized

[1] 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.

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

Figure 1: Ground truth references and mask examples out of the studied autonomous driving Waymo sequences [12]. 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 Neural Atlas Graphs (NAG) representation, each of the masked instances will be attributed to its own atlas node.

together.

We use the Adam [5] optimizer, with an initial learning rate of 0.001 in combination with a "ReduceLROnPlateau" scheduler, which will be activated from epoch 20 on. The neural fields within every node are parameterized by 5-layer MLPs (64 neurons, ReLU), while the input coordinates are encoded with a 16-level multi-resolution hash encoding [10] (4 features/level, scale 1.61, hashmap 217, base res. 4, linear interp.). We state the actual sizes in a dedicated section 2.5. For the Waymo [12] 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 [11] 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 affect of the control points is briefly studied within our ablations Sec. 2.4. 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`.

Baselines Within the autonomous driving scenes, we evaluate against OmniRe [1], a recent dynamic 3D Gaussian Splatting (3DGS) method, which was explicitly designed for autonomous driving scenes including a dedicated model for pedestrians and showing peak visual performance on the Waymo dataset. Further, given its architecture, it allows for positional edits of its modeled objects, but lacking support for texture editing. We also compare against EmerNeRF [14], 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. While OmniRe provides visualizations of positional editing functionalities, we could not find any implementations regarding this, given by the authors, so we amended the evaluation scripts to specify the target object IDs for manipulation, but we left the core model implementations untouched. For all methods we used all images of the datasets (as per experiment's subset division) to train the models, yielding a representation of maximal visual expressiveness.

2.3 Quantitative Results

As noted above, we benchmark our approach against the recent OmniRe [1] method—a 3DGS framework with explicit SMPL-based human modeling [8]—and EmerNeRF [14]. Tab. 2 lists the PSNR, SSIM [13], and LPIPS [15] scores, along with their standard deviations over all different sub-segments of individual sequences, for each scene.

We also isolate and assess the dynamic elements by dividing them into a rigid "Vehicle" category and a non-rigid "Human" category, enabling us to see how each class leverages our underlying rigid-motion model. Tab. 3 reports per-class PSNR and SSIM [13] results with accompanying standard deviations over the sub-segments, demonstrating substantial improvements over the strongest baseline. This shows that our gains stem not from improved background rendering, but from the high fidelity of our model in capturing even non-rigid motion.

Finally, to verify generalization, we test on diverse outdoor sequences from the DAVIS dataset [11]—a high-resolution (up to 1920 × 1080) benchmark commonly used by matting methods [4, 9, 6]. Following the selection of 15 sequences in [4, 6] featuring varied objects, complex backgrounds, and dynamic camera moves, we summarize our results in Tab. 4, including standard deviations for PSNR, SSIM [13], and LPIPS [15].

2.4 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 [12]. 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. 5. The top row of the table, labeled "Large", represents the performance of our full reference model as described in the main manuscript. We evaluated the impact of varying the model size ("Medium" and "Small") 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. 2, further illustrating the qualitative differences.

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.

Table 2: Quantitative Evaluation on Dynamic Driving Sequences of the Waymo [12] Open Driving Dataset with standard deviations calculated over sub-segments of individual sequences, and the mean standard deviation over each sequence. Best results are in bold. ORe refers to OmniRe [1], and ERF to EmerNeRF [14].

Seq.	PSNR ↑			SSIM ↑			LPIPS ↓		
	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF
s-975	**40.21** ±1.92	37.42 ±1.51	34.81 ±1.93	**0.976** ±0.005	0.969 ±0.005	0.941 ±0.011	**0.084** ±0.013	0.085 ±0.012	0.143 ±0.021
s-203	**43.15** ±0.77	36.99 ±0.54	35.15 ±0.37	**0.978** ±0.002	0.968 ±0.001	0.945 ±0.006	**0.070** ±0.006	0.097 ±0.004	0.205 ±0.017
s-125	**43.32** ±2.07	39.03 ±2.10	35.04 ±1.60	**0.980** ±0.006	0.971 ±0.004	0.937 ±0.012	**0.057** ±0.018	0.083 ±0.006	0.183 ±0.029
s-141	**42.55** ±1.04	36.23 ±1.10	34.83 ±2.13	**0.978** ±0.004	0.965 ±0.002	0.934 ±0.012	**0.057** ±0.014	0.090 ±0.003	0.171 ±0.035
s-952	**41.89** ±2.50	39.90 ±0.77	35.80 ±0.77	0.976 ±0.012	**0.978** ±0.004	0.942 ±0.012	0.058 ±0.028	**0.052** ±0.006	0.120 ±0.024
s-324	**40.85** ±0.59	32.67 ±2.51	33.68 ±0.92	**0.977** ±0.002	0.954 ±0.010	0.926 ±0.005	**0.038** ±0.004	0.072 ±0.013	0.124 ±0.007
s-344	**41.84** ±0.24	36.78 ±0.87	35.30 ±1.06	**0.983** ±0.000	0.974 ±0.001	0.947 ±0.008	**0.031** ±0.002	0.044 ±0.002	0.084 ±0.011
Mean	**41.85** ±1.15	36.92 ±2.31	34.93 ±0.65	**0.978** ±0.002	0.968 ±0.007	0.939 ±0.007	**0.055** ±0.016	0.072 ±0.019	0.141 ±0.041

Table 3: Quantitative Evaluation of Human and Vehicle Rendering on Waymo [12] Driving Sequences with standard deviations calculated over sub-segments of each sequence, and the mean standard deviation over each sequence.

Seq.	Vehicle PSNR ↑			Vehicle SSIM ↑			Human PSNR ↑			Human SSIM ↑		
	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF	Ours	ORe	ERF
s-975	**45.55** ±0.94	32.92 ±1.94	30.14 ±2.71	**0.988** ±0.001	0.940 ±0.022	0.867 ±0.035	**43.41** ±1.18	31.39 ±0.72	29.12 ±3.08	**0.981** ±0.002	0.913 ±0.026	0.858 ±0.057
s-203	**41.90** ±2.10	30.68 ±1.60	27.09 ±2.52	**0.986** ±0.005	0.934 ±0.001	0.817 ±0.045	**45.40** ±0.00	33.65 ±0.00	33.54 ±0.00	**0.986** ±0.000	0.938 ±0.000	0.920 ±0.000
s-125	**41.00** ±1.41	30.02 ±0.87	23.85 ±0.76	**0.989** ±0.004	0.933 ±0.003	0.663 ±0.026	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** ±2.13	32.02 ±2.00	27.94 ±2.17	**0.981** ±0.027	0.928 ±0.047	0.799 ±0.050	**44.22** ±1.28	34.41 ±4.15	29.38 ±2.35	**0.986** ±0.005	0.924 ±0.031	0.833 ±0.047
s-952	**40.94** ±0.42	31.16 ±0.84	26.04 ±0.77	**0.986** ±0.001	0.922 ±0.013	0.780 ±0.023	**40.45** ±0.85	29.90 ±0.66	26.32 ±0.81	**0.968** ±0.010	0.874 ±0.010	0.759 ±0.032
s-324	**41.71** ±0.83	31.66 ±0.65	28.85 ±1.28	**0.986** ±0.001	0.941 ±0.003	0.845 ±0.022	**44.13** ±1.71	31.03 ±3.98	27.02 ±2.16	**0.988** ±0.002	0.906 ±0.048	0.776 ±0.054
s-344	**43.97** ±1.16	28.31 ±1.04	30.45 ±1.33	**0.985** ±0.005	0.898 ±0.017	0.859 ±0.018	**40.99** ±1.00	28.84 ±0.57	27.51 ±2.08	**0.975** ±0.004	0.877 ±0.014	0.823 ±0.064
Mean	**42.69** ±2.09	31.01 ±1.51	27.94 ±2.34	**0.986** ±0.003	0.928 ±0.015	0.808 ±0.070	**42.89** ±2.17	31.40 ±2.14	28.38 ±2.60	**0.980** ±0.008	0.903 ±0.026	0.820 ±0.058

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.

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.

Table 4: Quantitative evaluation results on the Davis Dataset [11] of diverse outdoor scenes with standard deviation over all images from each scene. The best results are in bold for all metrics.

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

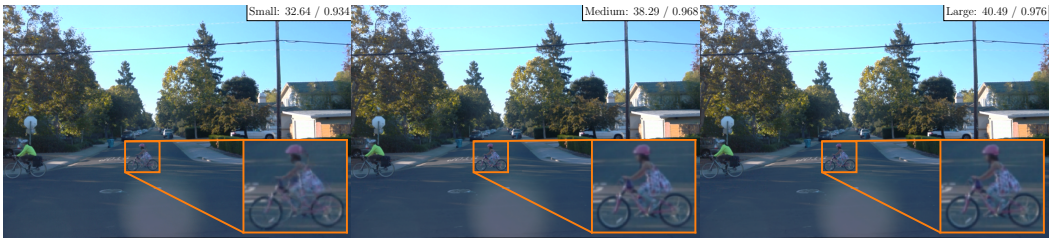

Figure 2: Representative examples of NAG nodes with varying parametrization sizes (Small, Medium, Large), including their PSNR / SSIM scores. Noticeable image quality degradation and flow collapsing artifacts are evident in the small node due to its limited representation, whereas distinguishing visual differences between medium and large nodes is challenging, with only minor lighting variations on the ground.

We also assessed the importance of the instance segmentation mask loss ("Excl. Mask-Loss") and the translation learning component ("Excl. Translation Learning"). While the impact of removing the mask loss appears relatively minor on the overall metrics, an exclusion will lead to much more opaque

Table 5: Ablation Experiments. We conducted ablation studies on a subset of our Waymo Datasets [12], 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. Mask-Loss	41.31	**0.977**	0.058	44.95	0.986	44.88	0.987
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

objects, the degradation observed when translation learning is excluded suggests its contribution to accurate object motion handling.

Finally, we explored the effect of explicitly learning plane rotations, similarly as we did within our DAVIS [11] 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 a more extensive evaluation on the full Waymo dataset, suggesting that its benefit might be scene-specific or less pronounced in more diverse scenarios.

The last 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.

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.

2.5 Parametrization

We state the parameterization of each node in our NAG in Tab. 6. 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 stuck to a single size for simplicity.

Table 6: 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

2.6 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 [11], including reconstructions and their corresponding scene decompositions, providing deeper insights into our model's performance and representation.

In Fig. 3 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. 4 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.

In Fig. 5, we demonstrate a realistic editing and seamless texture blending within the Waymo s-203 sequence. The edited car integrates naturally with the street, including the painted speed sign and zebra crossing. Crucially, the shadow casts realistically along these edited areas, showcasing our method's ability to maintain natural occlusion behavior. Furthermore, both the person in the foreground and the car in the background are removed without recognizable artifacts, highlighting the precision of our editing process.

Figure 6 showcases the visual effectiveness of our method on the DAVIS Dataset [11]. 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 7 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 [11] sequences in Fig. 8 and Fig. 9 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.

Figure 3: Extended visual results on the Waymo Dataset [12], 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 4: 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.

Figure 5: Effective scene manipulation in s-203. Four timestamps from the s-203 scene are shown, with ground truth (top) and our edits (bottom). The applied background texture is presented top-right, and the removed objects are shown bottom-right. The natural blending of the car and its shadow with the edits demonstrates our realistic occlusion handling.

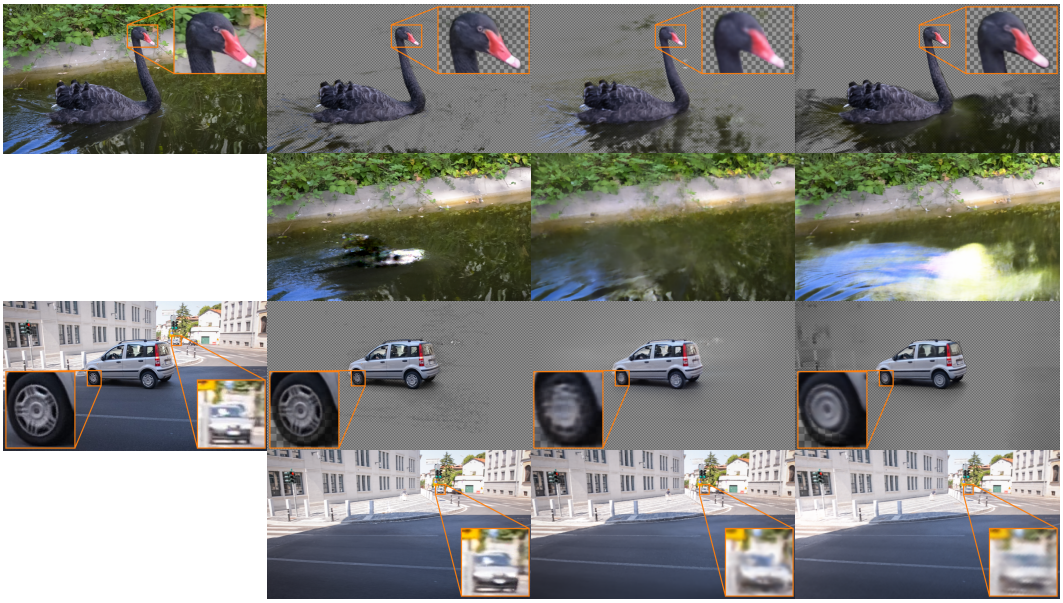

Figure 6: Visual comparison and decomposition of the blackswan and car-shadow sequence within the DAVIS dataset. The insets stating difficult regions underlining the capabilities of our model in accurately representing highly textured regions (swan head), time-variant content (spinning wheels) and distant background objects.

Figure 7: Additional visual examples of the DAVIS [11] sequences motorbike, bear and hike, showcasing our models quality in representing fine and complex details on rigid and non-rigid foreground actors.

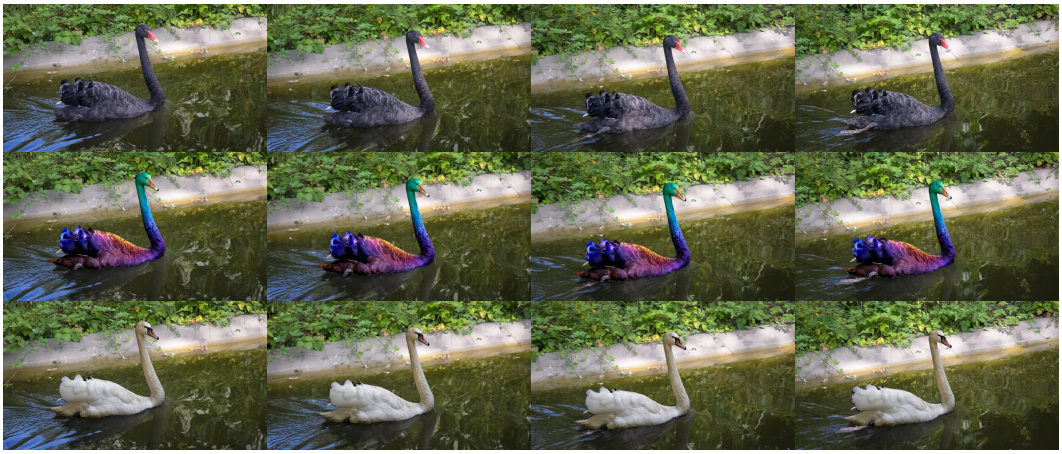

Figure 8: Advanced texture editing applied to the blackswan sequence (DAVIS dataset [11]). The top row presents the ground truth. The lower rows display edits where an off-the-shelf image generation model was leveraged to create rainbow and white swan texture variants. Using these generated textures, our method effectively propagates these localized changes consistently across all video frames, demonstrating robust temporal coherence.

Figure 9: Texture edits using DAVIS [11] boat sequence. Similar to Fig. 8, 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.

References

[1] 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.

[2] 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.

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.