Supplementary Materials: WavePlanes: Compact Hex Planes for Dynamic Novel View Synthesis

I. Introduction

The supplementary material is accompanied with rendered results and side-by-side video comparisons, which can also be accessed online.

In this document, we provide an overview of the differences in our hyper parameters compared to the K-Planes and 4D-GS models in Section II. In Section III we provide the per-scene quality and compression results. In Section IV, we provide additional experiments. For example we evaluate the effect of varying the hard threshold parameter. Finally, in Section V we discuss the paper's limitations.

II. Additional Hyper Parameters

All configuration files are provided with our code online. As the proposed work builds upon existing methods, we used the same hyper parameters as K-Planes and 4D-GS with minor differences:

1) For the synthetic D-NeRF data set: We tuned the weights for regularizing WavePlanes-NeRF per scene. For compression a hard threshold of 0.3 was selected. Additionally, for some synthetic dynamic scenes our model performs best with lower spatial resolutions as shown in Table I. Hence, scenes with finer details use $H = W$ =256, while scenes with lower frequency detail use $H = W$ =128[1]. The chosen resolution for each scene is provided in the configuration files in our code online.
2) For the real DyNeRF scenes [1] we use 8× down sampling to generate the IST weights and 2× down sampling to train the model. The main difference is that WavePlanes-NeRF uses feature length of 64, double the feature length used in K-Planes. This doesn't significantly increase computation however it does improve quality. For WavePlanes-GS we use a feature length of 32. Finally, for compression a hard threshold of 0.1 was selected.

Furthermore, the level-dependent wavelet coefficient scaling factor **k** was selected by testing the WavePlanes-NeRF model on the T-Rex D-NeRF scene in Table II. This factor was used for all experiments including the WavePlanes-GS model.

III. Per-Scene Results

Real dynamic scenes: In Table III we breakdown the results from each scene in the DyNeRF data set.

Synthetic dynamic scenes: In Table IV we breakdown and compare the results from the D-NeRF data set.

[1]These parameters were defined in the main paper

TABLE I: Varying the resolution of WavePlanes-NeRF feature grids for various D-NeRF scenes. The resolution of the resulting features planes is shown, where the wavelet coefficients will have a maximum resolution half that of P_c^0

Resolution P_c^0	Resolution P_c^1	PSNR ↑	SSIM ↑	Time ↓
T-Rex scene, D-NeRF [2]				
128	64	30.88	0.9749	62 mins
256	128	31.34	0.9782	72 mins
512	256	30.75	0.9761	127 mins
Lego, D-NeRF [2]				
128	64	25.25	0.9380	60 mins
256	128	25.19	0.9876	68 mins
512	256	24.76	0.9377	113 mins
Bouncing Balls, D-NeRF [2]				
128	64	37.71	0.9876	70 mins
256	128	36.66	0.9840	74 mins
512	256	34.74	0.9802	110 mins

TABLE II: **Level-dependent scaling coefficients, k**, used for the comparing different levels of wavelet decomposition of the WavePlanes-NeRF model. Accomplished on the T-Rex D-NeRF scene. Training time is provided in minutes. *Front* and *Back* indicate the PSNR applied to the foreground and background respectively, accomplished using morphological dilation on the alpha channel of the ground truth RGBA images

Scaling Factor, k	PSNR ↑ Whole	PSNR ↑ Front	PSNR ↑ Back	SSIM ↑
$N = 2$ levels, 72 mins				
$[1, 1, 1]$	31.30	20.63	76.01	0.977
$[1, 0.8, 0.4]$	31.31	20.64	75.78	0.978
$[1, 0.4, 0.2]$	31.34	20.67	78.05	0.978
$[1, 0.8, 0.2]$	31.33	20.66	76.75	0.978
$N = 3$ levels, 84 mins				
$[1, 1, 1, 1]$	30.52	19.53	75.39	0.975
$[1, 0.8, 0.6, 0.4]$	30.98	20.27	75.41	0.977
$[1, 0.5, 0.3, 0.1]$	30.64	19.97	78.00	0.975
$[1, 0.8, 0.4, 0.2]$	30.37	19.70	76.21	0.975
$N = 4$ levels, 95 mins				
$[1, 1, 1, 1]$	30.03	19.37	62.34	0.973
$[1, 0.5, 0.4, 0.2, 0.1]$	30.65	19.98	75.08	0.975
$[1, 0.8, 0.6, 0.4, 0.2]$	30.22	19.56	69.29	0.974

Per-scene compression results: In Table III and Table IV we provide the per-scene model size after compression. The proposed compression scheme performs optimally in the presence of empty space, i.e. when the wavelet coefficients are near-zero. This is best exemplified by the results from the D-NeRF scenes, where emptier scenes undergo higher compression.

TABLE III: **Quantitative results from the multi-view real DyNeRF dynamic scenes [1]**. * Uses 8x down sampling for IST weights. **Trained on the first 10 seconds of a 40 second clip

Method	Spinach	Cut Beef	Flame Salmon	Flame Steak	Sear Steak	Mean	**Size MB**
Per-Scene Model Size ↓							
K-Planes-Compact	69MB	72MB	84MB	65MB	66MB	71MB	-
Ours-NeRF	58MB	58MB	57MB	49MB	62MB	57MB	-
Ours-GS	33MB	34MB	49MB	32MB	32MB	36MB	-
PSNR ↑							
K-Planes	32.19	31.93	28.71 **	31.80	31.89	31.30	250
HexPlanes	32.04	32.55	29.47	32.08	32.09	31.65	200
4D-GS	32.460	32.90	29.20	32.51	32.49	31.91	90
K-Planes-Compact*	29.22	30.93	25.27 **	29.07	29.64	28.83	71
WavePlanes-NeRF*	31.04	31.45	28.25 **	30.49	30.37	30.32	57
WavePlanes-GS	31.97	32.63	28.90	32.10	32.34	31.59	36
SSIM ↑							
K-Planes	0.968	0.965	0.942 **	0.970	0.971	0.963	-
4D-GS	0.949	0.957	0.917	0.954	0.957	0.947	-
K-Planes-Compact	0.915	0.932	0.871	0.929	0.930	0.915	-
WavePlanes-NeRF*	0.9191	0.9338	0.8928	0.9364	0.9271	0.9218	-
WavePlanes-GS*	0.9434	0.9442	0.9106	0.9514	0.9525	0.9404	-

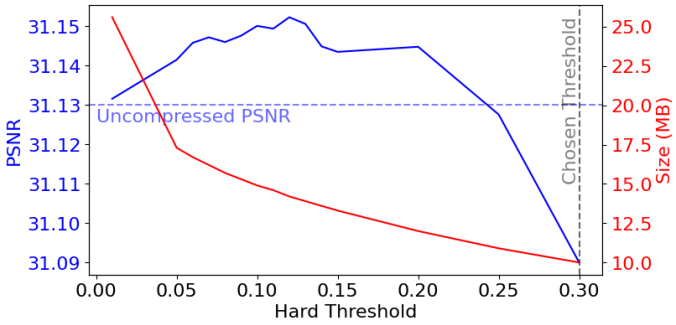


Fig. 1: Selecting a hard threshold: Quality deprecation as a result of high pass filtering is not proportional to decrease in model size as hard thresholding can act as a denoiser. We exemplify this on the T-Rex scene [2] where a theshold of 0.3 was selected.

IV. Additional Experiments

A. Selecting the Hard Threshold

We investigate the effect of varying the hard threshold parameter for compression in Fig. 1 on WavePlanes-NeRF and show that quality does not decrease proportionally to size. This demonstrates that our compression can provide minor denoising. This warrants further investigation.

B. Visualizing Feature Planes

The six feature planes for WavePlanes-NeRF, $\mathbf{P}_c^0$, are visualized in Fig. 2 for the D-NeRF T-Rex scene. This provides an example of how empty space and/or space time can consume a lot of data, however this also indicates that our compression algorithm will not perform as well for denser scenes. This is exemplified by comparing the WavePlanes-NeRF results on the D-NeRF (less dense) and DyNeRF (more dense) scenes in Table III and Table IV. Whereby, denser scenes require more memory.

C. Decomposing Static and Dynamic Components

In Fig. 3, we illustrate how our representation can be decomposed into static and dynamic components. To render a space-only scene (visualizing only static features), we force the condition $f_{c_t}(\mathbf{q}) = 1$ by zeroing the wavelet coefficients for all space-time planes. The space-only rendered frames are subtracted from the final render to visualize the effect of space-time features. Note that we do not use $f_{c \neq c_t}(\mathbf{q}) = 1$ to render space-time features directly as they can not be interpreted by the linear decoder. This adds to our interpretation of space-time features, whereby we treat them as linear transformations for the space-only features that can be interpreted as basis features. Interestingly, this behavior is similar to dynamic NVS that use deformation fields to linearly transform the position of volumes in a static field. Though instead, the plane representation only modifies the density and color of volumes in time.

V. Paper Limitations

A. Quantitative Image Assessment Metrics

It is challenging to discern true dynamic performance from the variety of metrics that have been proposed, such as SSIM, D-SSIM [3], MS-SSIM [4] and LPIPS [5]. For synthetic RGBA data sets we also used the alpha channel to separate foreground and background predictions during testing and validation. Considering that all these metrics evaluate stationary frames at different times, we are limited by our ability to evaluate temporal features such as smoothness and consistency. Additionally, the data sets we use do not support this type of evaluation. For instance, the D-NeRF data set is strictly provided as a set of "teleporting" frames so could

TABLE IV: **Quantitative results from the monocular synthetic D-NeRF dynamic scenes [2]**. N refers to the wavelet level (we select $N = 3$)

Method	Hell Warrior	Mutant	Hook	Balls	Lego	T-Rex	Stand Up	Jumping Jacks	Mean	**Size MB**
Per-Scene Model Size ↓										
Ours-NeRF	3.6MB	3.4MB	5.9MB	1.3MB	9.7MB	14.6MB	2.7MB	9.0MB	6.3MB	-
Ours-GS	13.8MB	16.3MB	15.3MB	9.9MB	33.4MB	26.9MB	10.3MB	9.0MB	16.9MB	-
PSNR ↑										
DNeRF	25.02	31.29	29.25	32.80	21.64	31.75	32.79	32.80	29.67	13
TiNeuVox-s	27.00	31.09	29.30	39.05	24.35	29.95	32.89	32.33	30.75	8
TiNeuVox-B	28.17	33.61	31.45	40.73	25.02	32.70	35.43	34.23	32.67	48
V4D	27.03	36.27	31.04	42.67	25.62	34.53	37.20	35.36	33.72	377
K-Planes	25.60	33.56	28.21	38.99	25.46	31.28	33.27	32.00	31.05	200
HexPlanes	24.24	33.79	28.71	39.69	25.22	30.67	34.36	31.65	31.04	200
4D-GS	28.71	37.59	32.73	40.63	25.03	34.23	38.11	35.42	34.06	18
Ours-NeRF	25.85	33.25	27.77	38.42	25.31	31.46	33.27	31.87	30.90	6.3
Our-GS	29.05	38.71	33.40	40.45	25.02	35.88	39.10	34.87	34.56	16.9
SSIM ↑										
DNeRF	0.95	0.97	0.96	0.98	0.83	0.97	0.98	0.98	0.95	-
TiNeuVox-s	0.95	0.96	0.95	0.99	0.88	0.96	0.98	0.97	0.96	-
TiNeuVox-B	0.97	0.98	0.97	0.99	0.92	0.98	0.99	0.99	0.97	-
V4D	0.97	0.98	0.97	0.99	0.92	0.98	0.99	0.98	0.97	-
K-Planes	0.951	0.982	0.951	0.989	0.947	0.980	0.980	0.974	0.969	-
HexPlanes	0.94	0.98	0.96	0.99	0.94	0.98	0.98	0.98	0.97	-
4D-GS	0.9733	0.9880	0.9760	0.9943	0.9376	0.9850	0.9898	0.9857	0.9787	-
Ours-NeRF	0.9536	0.9775	0.9461	0.9880	0.9428	0.9786	0.9779	0.9734	0.9672	-
Ours-GS	0.9752	0.9906	0.9783	0.9942	0.9393	0.9878	0.9913	0.9848	0.9802	-

not be used for evaluating spatiotemporal smoothness using ground truth renders.

B. Hardware Failure Case

Our work began with 24 GB of GPU memory and 32 GB of RAM. This works for the D-NeRF set. However, for the DyNeRF data set the amount of RAM required for IST weight generation is significant (>256GB). We were unable to attain this during our research. Instead, we found that 98 GB of RAM was sufficient and low cost, despite limiting us to $\times 8$ down sampling for IST weight generation.

VI. Failed Designs

The final design was not the first solution we conceived. In this section we detail several technical designs that failed.

A. K-Planes Time Smoothness Regularizer

As our implementation initially forked off the K-Planes code, our earliest design utilized the TS regularizer proposed in [6] instead of the proposed SST regularizer. In Table V we compare the results from using TS regularization and SST regularization, showing that for WavePlanes-NeRF the SST regularizer is a better fit.

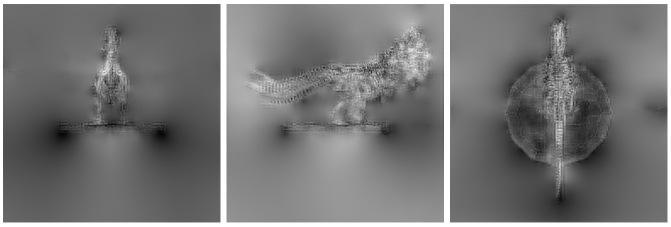
$$\mathcal{L}_{TV} = \frac{1}{|C|n^2} \sum_{c,i,t} ||\mathbf{P}_c^{i,t-1} - 2\mathbf{P}_c^{i,t} + \mathbf{P}_c^{i,t+1}||_2^2 \qquad (1)$$

TABLE V: **Comparing SST and TS regularizers** on the T-Rex D-NeRF scene [2]

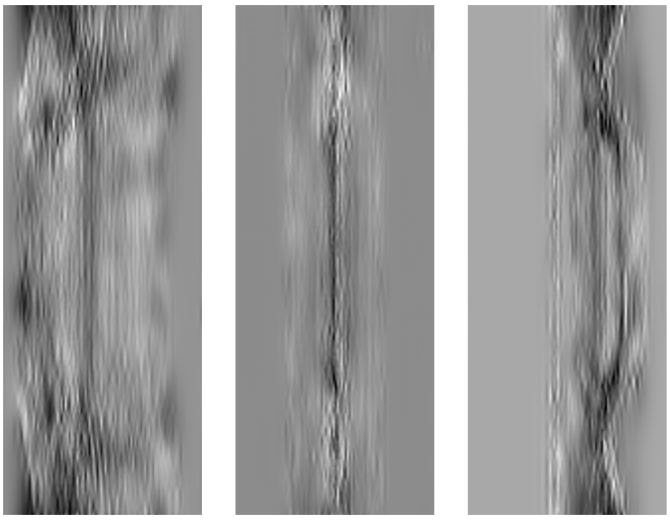
Regularizer	PSNR ↑ Front	Back	Whole	SSIM ↑
SST	20.74	76.41	31.41	0.9781
TS	18.70	75.29	29.37	0.9721

B. Smoothing Temporal Coefficients with Wavelet Filter Orientation

Each set of mother wavelet coefficients contains components for horizontal, diagonal and vertical filters which we define as $\Omega_{c,\omega} \in \mathbf{\Omega}_c$ where $\omega \in [horizontal, vertical, diagonal]$. This offers the opportunity to refactor the SST function (a 1-d Laplacian approximation of the planes second derivative) to regularize coefficient filters directly. This also allows us to prioritize smoothness for each filter direction, where the horizontal filter exists along the time-axis, the vertical filter along the spatial axis and the diagonal filter along $t = x = z = y$. Hence we use (2) for each filter where $c = c_t$. To regularize the father wavelet coefficients, which do not contain directional filters, we apply the 1-D Laplacian approximation across the horizontal and vertical axis and add the result to $\mathcal{L}_{SST-horizontal}$ and $\mathcal{L}_{SST-vertical}$, respectively. For $\omega = diagonal$, we average the second order approximations along both axis' of the father wavelet planes and add the result to $\mathcal{L}_{SST-diagonal}$. This ensures that all coefficients are regularized for a given orientation. The result of using these separately is compared with the proposed SST



(a) Space-only Feature Planes



(b) Space-time Feature Planes

Fig. 2: Visualizing the space-only and space-time feature planes for WavePlanes-NeRF. Black to white pixels indicate negative to positive feature values, respectively. (a) Space-only feature planes are visualized. (b) Space-time features are visualized where the horizontal axis represents time

TABLE VI: **Comparing directionally dependent smoothness regularizers** on the wavelet coefficients for the T-Rex D-NeRF data set [2]. * indicates the final Wave Planes model which uses the proposed SST regularization

ω	PSNR ↑	
	Whole	Front
*	31.34	20.67
horizontal	29.23	18.55
diagonal	29.14	18.41
vertical	29.08	18.48

regularization in Table VI for WavePlanes-NeRF.

$$\mathcal{L}_{SST-\omega} = \frac{1}{|C|n^2} \sum_{c,i,t} ||\mathbf{\Omega}_{c,\omega}^{i,t-1} - 2\mathbf{\Omega}_{c,\omega}^{i,t} + \mathbf{\Omega}_{c,\omega}^{i,t+1}||_2^2 \quad (2)$$

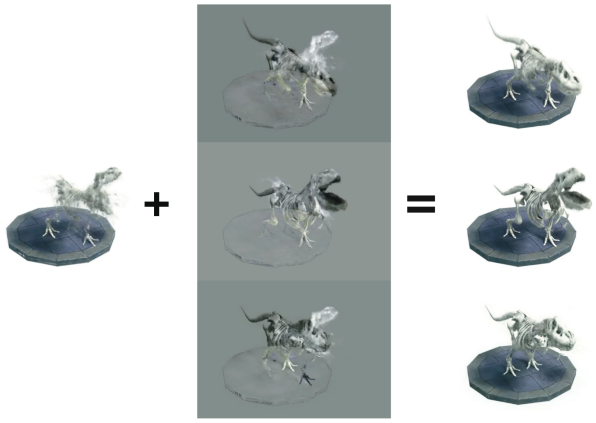


Fig. 3: Visualizing space and space-time separation for WavePlanes-NeRF on the T-Rex D-NeRF scene. **Left**: Space-only features are rendered. **Right**: All features are rendered. **Center**: All rendered features are subtracted from the space-only render, representing the effect of space-time features on the final render

REFERENCES

[1] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, et al., "Neural 3d video synthesis from multi-view video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5521–5531. 1, 2

[2] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10318–10327. 1, 2, 3, 4

[3] Allison H Baker, Alexander Pinard, and Dorit M Hammerling, "Dssim: a structural similarity index for floating-point data," *arXiv preprint arXiv:2202.02616*, 2022. 2

[4] Zhou Wang, Eero P Simoncelli, and Alan C Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Ieee, 2003, vol. 2, pp. 1398–1402. 2

[5] 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. 2

[6] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12479–12488. 3