

From NeRFs to Gaussian Splats, and Back

Siming He*, Zach Osman*, Pratik Chaudhari

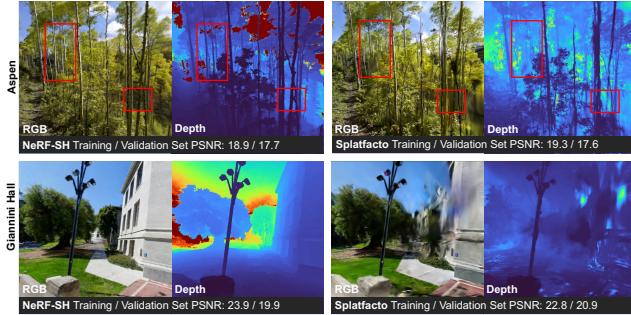


Figure 1. NeRF generalizes better than Gaussian splatting (GS) to views that are very different from those in the training data. NeRF-SH (Left) and Splatfacto (Right), both summarized in Tab. 1, are trained on Aspen (Top) and Giannini Hall (Bottom) from Nerfstudio dataset [1]. In both datasets, NeRF-SH and Splatfacto have relatively good training and validation PSNRs to one another because the validation views are similar to the training views (see Fig. 2 Top). For novel views which differ more from the training views, like the images shown above, NeRF-SH renders better RGB and depth images than Splatfacto. The red boxes in the Aspen novel view illustrate areas in the RGB and depth views where NeRF-SH has noticeably better depth geometry and fewer artifacts than Splatfacto. For the Giannini Hall novel view, NeRF-SH clearly preserves the depth structure better than Splatfacto.



Figure 2. Top: Aspen and Giannini Hall scenes from Nerfstudio [1] contain validation views that are similar to training views. Although Splatfacto has good rendering quality on these data in Tab. 2, it does not accurately render dissimilar views in Fig. 1. Bottom: We created two new scenes, named Wissahickon and Locust Walk, where validation views are much more different than training views. For these new scenes, Splatfacto has a much worse validation PSNR than NeRF; see Tab. 2. All images above are rendered using Gaussian splats obtained from NeRF-SH without fine-tuning; this is called NeRFGS with zero iterations in Tab. 2.

Radiance field-based scene representations are useful in robotics for localization and mapping [2–8], planning and control [9–13], scene understanding [14, 15], and simulation [16–18]. Often, the key question in these applications is whether one uses an implicit representation like a neural

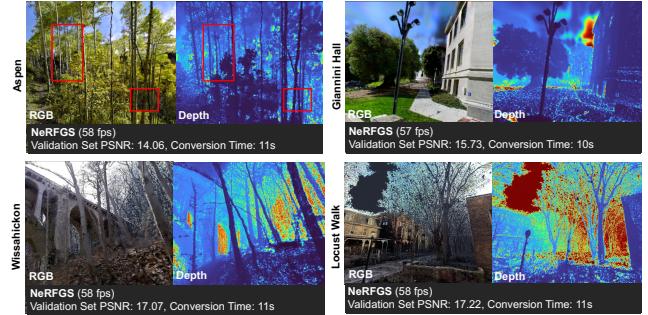


Figure 3. NeRFGS generalizes better than GS while having real-time rendering. NeRFGS converts trained NeRF-SH into GS while maintaining good generalization in contrast to Splatfacto in Fig. 1. The conversion takes about 10 sec on GeForce RTX 4090: 7 sec for extracting spherical harmonics and 3 sec for fine-tuning. It is therefore fast enough to be done periodically on a robot. If necessary, this time can be reduced if NeRF-SH to GS conversion is done only around the robot; or if the sky (not relevant for many ground robotics tasks) is ignored.

Nerfacto	NeRF-based approach in Nerfstudio
NeRF-SH	Our modified Nerfacto that predicts spherical harmonics for the color instead of the RGB intensity
Splatfacto	Gaussian Splatting approach in Nerfstudio
NeRFGS	Gaussian splats obtained from NeRF-SH, with or without further fine-tuning
GSNeRF	NeRF-SH converted from NeRFGS
RadGS	Gaussian Splatting trained using the pointcloud obtained from NeRF [23]

Table 1. Different approaches for scene representations.

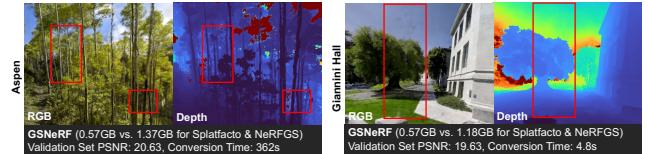


Figure 4. GSNeRF can be used for saving memory and editing the NeRF. We convert NeRFGS model, as in Fig. 3, back to NeRF-SH to get GSNeRF (Left). GSNeRF stores memory intensive GS models as more compact NeRFs. Also, GSNeRF with NeRFGS enables easy modifications to the scene represented by the NeRF by converting into a GS, modifying the Gaussians, and converting it back. As an example, we have edited out the lamp post from Giannini Hall in Fig. 1 through GSNeRF and rendered the image (Right) using NeRF-SH.

radiance field (NeRF) [19, 20] or an explicit representation, like 3D Gaussian Splatting (GS) [21, 22]. There are pros and cons for both.

Imagine a quadruped robot walking along a road. It is important to ensure that the scene representation built from its ego-centric views generalizes to new views from the other side of the road. Non-parametric GS models perform well when train are plenty and test views are similar to train views [19, 24]. Parametric models like NeRFs work much better when train and test views are different from each other. See Fig. 1. Heuristics to move, merge, and split the Gaussians in GS are brittle for in-the-wild data with exposure variations and motion blur [23]. Training of NeRFs is more stable and

*Equal Contribution. General Robotics, Automation, Sensing and Perception (GRASP) Laboratory, University of Pennsylvania. Email: {siminge, osmanz, pratikac}@seas.upenn.edu

Code and data are in <https://github.com/grasp-lyrl/NeRFtoGSandBack>.

Iterations ($\times 10^2$)	Aspen				Giannini Hall				Wissahickon				Locust Walk			
	PSNR (Val)	SSIM	LPIPS	PSNR (Val)	SSIM	LPIPS	PSNR (Train/Val)	SSIM	LPIPS	PSNR (Train/Val)	SSIM	LPIPS	PSNR (Train/Val)	SSIM	LPIPS	
Nerfacto-big [1]	300	17.75	0.5	0.43	20.11	0.68	0.3	22.17 / 20.75	0.75	0.26	22.29 / 21.49	0.8	0.3			
Splatfacto [1]	300	17.63	0.5	0.39	20.87	0.7	0.33	23.46 / 14.62	0.55	0.45	24.04 / 17.72	0.7	0.31			
NeRF-SH	300	17.73	0.48	0.45	19.89	0.65	0.32	22.41 / 17.46	0.61	0.39	21.73 / 18.74	0.7	0.33			
RadGS [23]	1	11.65	0.28	0.74	12.37	0.49	0.61	12.4 / 15.17	0.62	0.46	10.84 / 11.85	0.6	0.46			
RadGS [23]	10	17.85	0.51	0.44	20.84	0.72	0.3	20.7 / 20.73	0.76	0.29	21.15 / 21.04	0.8	0.25			
NeRFGS	0	13.96	0.3	0.58	16.19	0.47	0.49	- / 14.40	0.47	0.51	- / 14.87	0.51	0.47			
NeRFGS	1	14.06	0.34	0.57	15.73	0.53	0.46	16.62 / 17.07	0.63	0.4	15.7 / 17.22	0.65	0.37			
NeRFGS	10	17.7	0.51	0.4	21.05	0.73	0.26	20.67 / 20.64	0.75	0.27	21.11 / 21.14	0.8	0.24			
GSNeRF	50	18.1	0.44	0.44	21.22	0.69	0.31	- / 17.65	0.63	0.39	- / 19.32	0.71	0.33			
GSNeRF	300	18.58	0.51	0.36	23.71	0.82	0.17	- / 17.59	0.64	0.37	- / 19.32	0.72	0.31			

Table 2. Quantitative Results. We calculate PSNR, SSIM, LPIPS for different models and scenes. For Wissahickon and Locust Walk, where validation views are dissimilar to training views, we additionally show the training PSNR since we notice a larger gap between the training and validation PSNR for Splatfacto compared to other methods. In contrast, the rendering quality at validation views is already reasonable for NeRFGS after 100 iterations of fine-tuning. After training for 1000 iterations, NeRFGS has better quality than Splatfacto and NeRF-SH. Compared to RadGS, the convergence is faster, i.e., better results after 100 iterations, also shown in Fig. 5. GSNeRF also trains much faster and results in better quality, see Fig. 5. All GS based methods can render at more than 40 FPS on GeForce RTX 4090. Training for 100 iterations takes about 6 and 3 seconds, respectively, for NeRF-based and GS-based methods. GSNeRF was supervised using the Gaussians obtained from NeRFGS with 5000 iterations.

recovers better geometry with limited views. NeRFs are also a more compact representation and require less memory than GS. This is important for resource-constrained robots. The difference is rather obvious for distilled feature fields. NeRF-based methods [14, 25] can store high-dimensional features efficiently. GS-based methods [26–28] need additional steps to compress features.

Explicit representations can achieve faster rendering than implicit ones. High-speed rendering is important in robotics for localization (which requires checking many views to ascertain visual overlap with the current observation), planning (which requires synthesizing new views along putative trajectories), etc. Explicit representations can also be modified easily, e.g., by updating the Gaussians. This is useful for robots that operate in dynamic environments. Modifying implicit representations requires expensive re-training or complex modeling [29–33].

We develop a procedure to go back and forth between implicit and explicit representations. We evaluate the quality and efficiency of this approach using a number of existing datasets. We study this approach on views recorded from an ego-centric camera along hiking trails in situations when evaluation views are dissimilar to training views. We show that our approach achieves the best of both NeRFs (superior PSNR, SSIM, and LPIPS on the dissimilar views, and a compact representation) and GS (real-time rendering and ability for easily modifying the representation). The computational cost of converting between these representations is minor compared to training from scratch.

Results Tab. 1 provides a brief summary of the the different approaches. We modify Nerfacto to predict spherical harmonics (degree 3, i.e., 16 coefficients) for each RGB channel. The volume rendering equation remains unchanged: we calculate the RGB color from spherical harmonics using the viewing direction before integrating it along the ray.

Given such a trained “NeRF-SH”, we calculate a point-cloud of the scene using the median depth along 2×10^6 rays rendered from training views. We ensure that these rays have high opacity and do not correspond to the sky.

Isotropic Gaussians are initialized at each of these points using the density and spherical harmonics predicted by the NeRF-SH. The scale of each Gaussian is half of the average distance between each point and its three nearest neighbors. Without any further optimization, this “NeRFGS” already captures geometric and photometric properties of the scene impressively well; see Fig. 2 and Tab. 2. We can fine-tune it further using training views; see Fig. 3 and Tab. 2.

For GSNeRF, we render images using NeRFGS from training views, and fit or update a NeRF-SH. We noticed that training NeRFs using GS-rendered views gives better PSNR, SSIM, and LPIPS as compared to using the original images; see Fig. 5 and Tab. 2. This is perhaps due to the absence of high-frequency structures in the GS-rendered views. One might also be interested in converting an explicit representation back into an implicit one. We show an example in Fig. 4 where we manually edit out the lamp-post by selecting the corresponding splats in NeRFGS and updating the NeRF through GSNeRF in 4.8 sec.

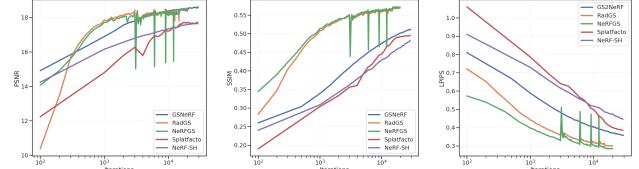


Figure 5. NeRFs can be efficiently converted to high-quality Gaussian splats. We report the PSNR, SSIM and LPIPS on validation data as a function of training progress for Aspen. After 1000 iterations of fine-tuning, NeRFGS performs comparably or better than NeRF-SH and Splatfacto.

Discussion We demonstrated a simple procedure to convert between implicit representations of the scene such as NeRFs and explicit representations such as Gaussian splatting (GS). These ideas are useful to handle situations with sparse views, which are commonly encountered in robotics. There are many ways one might build upon this work. Notice that in Tab. 2 the PSNR of NeRFGS without fine-tuning is lower than that of NeRF-SH. This indicates that there is a large degree of inefficiency in how we convert NeRF-SH into the explicit representation.

References

- [1] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023. [1](#), [2](#)
- [2] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6229–6238, October 2021. [1](#)
- [3] Antoni Rosinol, John J. Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444, 2023. doi: 10.1109/IROS55552.2023.10341922.
- [4] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photo-realistic mapping for monocular, stereo, and rgb-d cameras, 2024.
- [5] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R. Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting, 2023.
- [6] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam, 2024.
- [7] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting slam, 2024.
- [8] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting, 2024. [1](#)
- [9] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2): 4606–4613, 2022. doi: 10.1109/LRA.2022.3150497. [1](#)
- [10] Siming He, Christopher D. Hsu, Dexter Ong, Yifei Simon Shao, and Pratik Chaudhari. Active perception using neural radiance fields, 2024.
- [11] Guangyi Liu, Wen Jiang, Boshu Lei, Vivek Pandey, Kostas Daniilidis, and Nader Motee. Beyond uncertainty: Risk-aware active view acquisition for safe robot navigation and 3d scene understanding with fisherrf, 2024.
- [12] Xiaohan Lei, Min Wang, Wengang Zhou, and Houqiang Li. Gaussnav: Gaussian splatting for visual navigation, 2024.
- [13] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control, 2021. [1](#)
- [14] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *7th Annual Conference on Robot Learning*, 2023. [1](#), [2](#)
- [15] 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, 2023. [1](#)
- [16] Arunkumar Byravan, Jan Humplík, Leonard Hasenclever, Arthur Brussee, Francesco Nori, Tuomas Haarnoja, Ben Moran, Steven Bohez, Fereshteh Sadeghi, Bojan Vujatovic, and Nicolas Heess. Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields, 2022. [1](#)
- [17] Simon Le Cleac'h, Hong-Xing Yu, Michelle Guo, Taylor A. Howell, Ruohan Gao, Jiajun Wu, Zachary Manchester, and Mac Schwager. Differentiable physics simulation of dynamics-augmented neural objects, 2023.
- [18] Xiaohao Xu, Tianyi Zhang, Sibo Wang, Xiang Li, Yongqi Chen, Ye Li, Bhiksha Raj, Matthew Johnson-Roberson, and Xiaonan Huang. Customizable perturbation synthesis for robust slam benchmarking, 2024. [1](#)
- [19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. [1](#)
- [20] 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, July 2022. doi: 10.1145/3528223.3530127. [1](#)
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>. [1](#)
- [22] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields, 2024. [1](#)
- [23] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radspat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. *arXiv.org*, 2024. [1](#), [2](#)
- [24] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. [1](#)
- [25] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. [2](#)
- [26] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. *arXiv preprint arXiv:2311.18482*, 2023. [2](#)
- [27] Minghan Qin, Wanhu Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023.
- [28] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding, 2024. [2](#)
- [29] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. [2](#)
- [30] 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*, 2020.
- [31] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes, 2021.

- [32] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022.
- [33] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [2](#)