

Grow with the Flow: Modeling Plant Morphogenesis with 3D Gaussian Flow Fields

Supplementary Material

642 **A. Results on Real Captures**

643 Our captured dataset features a blooming flower recorded
644 using a Raspberry Pi equipped with an HQ camera module [38]. To ensure consistent multi-view imaging over
645 time, the plant is placed on a motorized turntable (lazy
646 Susan) controlled by the Pi. At each timestep, we capture
647 50 images at a fixed elevation with 7.2° angular spacing,
648 achieving full 360° coverage. Images are acquired every
649 15 minutes across 86 timesteps, documenting the full
650 blooming process. For training and evaluation, we use 43
651 views for training and 7 held-out views for testing at each
652 timestep. For training, we subsample 6 timesteps (selecting
653 every 17th frame), while all 86 timesteps—including the 80
654 temporally interpolated frames unseen during training—are
655 used for evaluation. Since the plant is relatively small com-
656 pared to its background, we compute masked PSNR, but
657 keep SSIM [41] and LPIPS [46] on the full image. The im-
658 age resolution of the captured data is 1200×1200 , which we
659 keep for training and evaluation. Camera poses and sparse
660 point clouds are computed using COLMAP [36].

661 In Figure 7, we show novel view renders of our method
662 versus baselines across 4 timesteps, where yellow bor-
663 ders indicate interpolated timesteps and no borders indicate
664 training timesteps. We observe that while baseline methods
665 have no difficulty reconstructing timesteps they have trained
666 on, their quality quickly degrades when queried on inter-
667 polated timesteps. Notably, 4D-GS completely fails during
668 interpolation where the reconstructed plant wobbles be-
669 tween growing and shrinking, despite the ground-truth motion
670 being a smooth growth motion, which is achieved by
671 our novel representation and training algorithm. In Figure
672 8, we show the interpolated point cloud trajectories. Again,
673 the recovered geometry of our method, especially at inter-
674 polated timesteps, appears superior compared to baselines.
675 In Table 2, we show our method’s image metrics across
676 training times, interpolation times, and combined. Despite
677 achieving slightly lower PSNR on training times, our SSIM
678 and LPIPS are both comparable to other methods. As our
679 neural ODE optimizes for a continuous flow field of Gaus-
680 sian parameters rather than overfitting to individual training
681 timesteps, it trades slightly lower performance on training
682 times for significantly superior interpolation quality, result-
683 ing in better overall combined performance.

684 We show more results of this scene in the Supp. Web-
685 page. We find that GROWFLOW enables rendering novel
686 views at held-out viewpoints and timesteps that are more
687 consistent with ground-truth captured data than baselines.

688 **B. GROWFLOW Training Algorithm** 689
689 We begin with a detailed outline of the training algorithm
690 of our pipeline in Algorithm 1. The first phase is the static
691 reconstruction stage, where we optimize a set of 3D Gaus-
692 sians on posed images of the fully grown plant. By the end,
693 we have optimized a set of Gaussians at timestep t_0 , which
694 we denote as \mathbf{G}^{t_0} . For the subsequent training phases, we
695 freeze color c and opacity o . Next, for the boundary recon-
696 struction, we integrate backwards in time, one timestep at a
697 time and cache the optimized Gaussians for each timestep.
698 After this phase, we have a set of cached Gaussians for each
699 timestep. Finally, during the global optimization step, we
700 randomly sample a timestep, and leverage the cached Gaus-
701 sian at that timestep to optimize the neural ODE. The result
702 is a trained neural ODE F_ϕ able to interpolate over unseen
703 timepoints. 704

705 **C. Implementation Details** 705
706 **Network implementation.** In this section, we provide a
707 detailed description of the network architecture. We imple-
708 ment our dynamic Gaussian representation using the open-
709 sourced Gaussian Splatting implementation gsplat [45] and
710 the neural ODE codebase torchdiffeq [5]. Our HexPlane
711 architecture follows closely [3, 42], where the spatial res-
712 olutions are set to 64 and the temporal resolution is set to
713 25, which are upsampled by 2. The learning rate of the
714 HexPlane is set to 1.6e-3, and the learning rate of the MLP
715 decoder is set to 1.6e-4, both of which are exponentially de-
716 cayed by a factor of 0.1 until the end of training, for 30K
717 iterations. Unlike [42], we omit the total variation loss, as
718 it does not bring additional improvement. We use a batch
719 size of 30 viewpoints for both our boundary reconstruction
720 stage and dynamic optimization stage, but keep the tempo-
721 ral batch size to 1. The MLP decoders consist of a two-layer
722 MLP with 64 units and a ReLU activation function. 722

723 For the synthetic experiments, after the static reconstruc-
724 tion, we fixed the background Gaussians and only input the
725 foreground Gaussians using a manually designed bounding
726 box. Doing so constrains the neural ODE to predicting the
727 flow field of foreground Gaussians, greatly easing optimiza-
728 tion.

729 **D. Additional Results** 729
730 Tables 3, 4, 5, 6 provide a breakdown of the quantitative
731 results in simulation across all scenes. Overall, our method
732 achieves state-of-the-art performance across all scenes com-

Table 2. Results for *Rose* scene assessing image quality across training, interpolation, and combined frames.

Method	Training times			Interpolation times			Combined		
	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓
4D-GS	30.14	0.991	0.025	22.58	0.983	0.032	23.11	0.983	0.032
4DGS	31.13	0.993	0.021	26.55	0.983	0.035	26.87	0.984	0.034
Dynamic 3DGS	28.86	0.989	0.027	24.99	0.975	0.045	25.26	0.976	0.044
Proposed	28.17	0.992	0.019	26.93	0.990	0.020	27.01	0.991	0.020

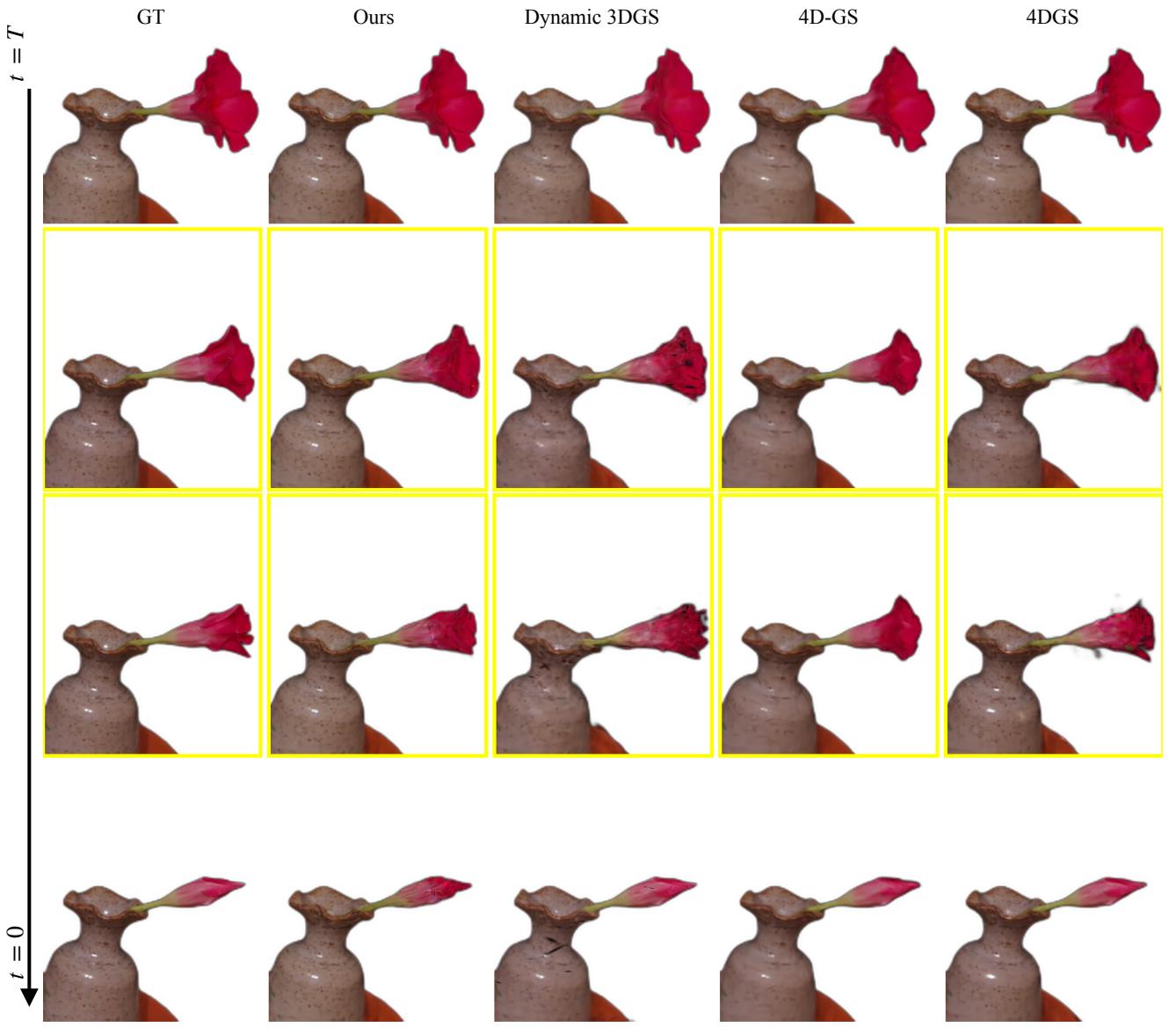


Figure 7. We show our method’s novel view renders against baselines on trained and interpolated timesteps. Our method reconstructs more faithfully the rose at interpolated timesteps compared to baselines (images indicated with a yellow border are novel view renders of interpolated times).

pared to baselines. Please refer to the Supp. Webpage for additional video results and comparisons to baselines for

simulated and captured results.

735

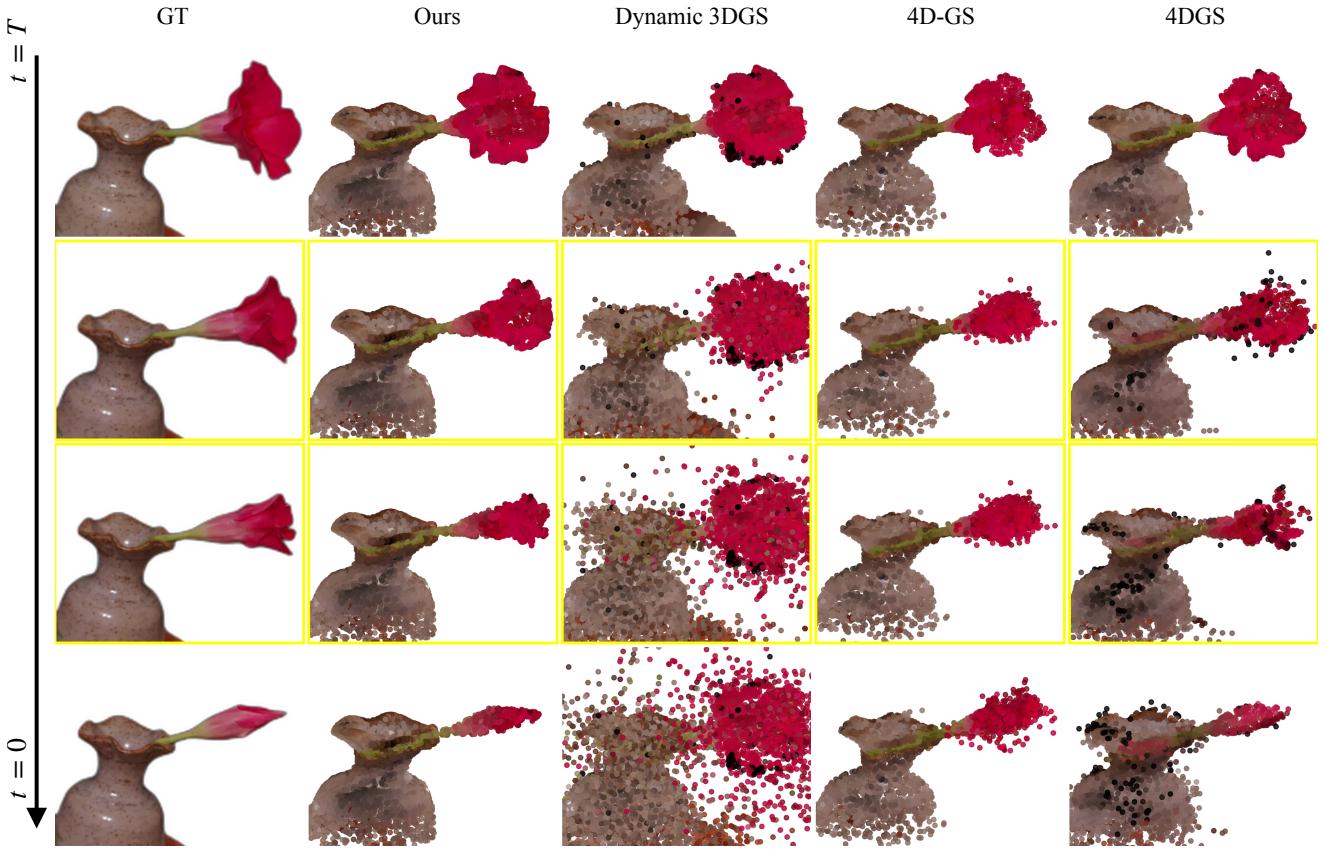


Figure 8. We show our method’s point cloud trajectories against baselines on trained and interpolated timesteps. Throughout the entire duration of growth, our method’s reconstructed geometry remains more faithful to the captured images.

Table 3. PSNR (dB) results across different synthetic scenes for combined (training + interpolation) frames.

Method	Clematis	Plant1	Plant2	Plant3	Plant4	Plant5	Tulip	Average
4D-GS	31.10	34.11	33.11	32.98	34.30	32.16	31.94	32.81
4DGS	27.62	29.78	29.24	29.73	30.06	29.50	29.13	29.29
Dynamic 3DGS	30.56	33.64	31.46	32.64	33.80	31.58	31.07	32.11
Proposed	33.40	37.34	32.50	35.15	36.84	32.99	34.55	34.68

Table 4. SSIM results across different synthetic scenes for combined (training + interpolation) frames.

Method	Clematis	Plant1	Plant2	Plant3	Plant4	Plant5	Tulip	Average
4D-GS	0.933	0.952	0.948	0.946	0.951	0.942	0.939	0.944
4DGS	0.887	0.922	0.911	0.914	0.921	0.910	0.908	0.910
Dynamic 3DGS	0.900	0.922	0.903	0.913	0.920	0.905	0.901	0.909
Proposed	0.951	0.967	0.943	0.962	0.965	0.940	0.962	0.956

Table 5. LPIPS results across different synthetic scenes for combined (training + interpolation) frames.

Method	Clematis	Plant1	Plant2	Plant3	Plant4	Plant5	Tulip	Average
4D-GS	0.102	0.087	0.095	0.095	0.089	0.097	0.095	0.094
4DGS	0.158	0.129	0.139	0.136	0.130	0.140	0.135	0.138
Dynamic 3DGS	0.162	0.148	0.165	0.156	0.152	0.161	0.155	0.157
Proposed	0.065	0.051	0.081	0.060	0.055	0.088	0.054	0.065

Table 6. CD results across different synthetic scenes for combined (training + interpolation) frames.

Method	Clematis	Plant1	Plant2	Plant3	Plant4	Plant5	Tulip	Average
4D-GS	0.21	0.20	2.03	0.22	0.17	2.42	0.12	0.77
4DGS	42.63	3.98	2.82	14.25	2.78	10.56	6.72	11.96
Dynamic 3DGS	79.26	0.79	2.32	1.98	0.22	0.40	9.98	13.56
Proposed	0.02	0.08	0.10	0.28	0.11	0.12	0.02	0.11

Algorithm 1 Training Loop for GROWFLOW

- 1: **Input:** Gaussians \mathbf{G} , posed images I_p^t , neural ODE F_ϕ .
- 2: **Parameters:** $n_{\text{static}} = 30000$, $n_{\text{boundary}} = 200$, $n_{\text{global}} = 30000$.
- 3:
- 4: **Step 1: Static Reconstruction**
- 5: **for** $epoch = 1$ to n_{static} **do**
- 6: Pick last timestep ground truth image $I_{\text{last}} = I_p^T$
- 7: $I_{\text{pred}} \leftarrow \text{Rasterize}(\mathbf{G})$
- 8: Compute $L \leftarrow \text{loss}(I_{\text{pred}}, I_{\text{last}})$
- 9: Update \mathbf{G}
- 10: **end for**
- 11: **Output:** $\mathbf{G}^{t_0} = (\mu^{t_0}, q^{t_0}, s^{t_0}, c, o)$
- 12:
- 13: **Step 2: Boundary Reconstruction**
- 14: **for** $k \in \{0, \dots, N - 1\}$ **do** ▷ Backwards in time
- 15: **for** $epoch = 1$ to n_{boundary} **do**
- 16: Pick ground truth image $I^{t_{k+1}}$
- 17: $\mathbf{G}^{t_{k+1}} = \mathbf{G}^{t_k} + \int_{t_k}^{t_{k+1}} F_\phi(\mu(t), t) dt$
- 18: $I_{\text{pred}} \leftarrow \text{Rasterize}(\mathbf{G}^{t_{k+1}})$
- 19: Compute $L \leftarrow \text{loss}(I_{\text{pred}}, I^{t_{k+1}})$
- 20: Update F_ϕ
- 21: **end for**
- 22: Cache $\mathbf{G}^{t_{k+1}}$
- 23: **end for**
- 24: **Output:** a set of cached Gaussians for each timestep
 $\{\mathbf{G}^{t_k}\}_k$
- 25:
- 26: **Step 3: Global Optimization**
- 27: Re-initialize new F_ϕ
- 28: **for** $epoch = 1$ to n_{global} **do**
- 29: Randomly sample timestep t_k
- 30: Pick ground truth image $I^{t_{k+1}}$
- 31: $\tilde{\mathbf{G}}^{t_{k+1}} = \mathbf{G}^{t_k} + \int_{t_k}^{t_{k+1}} F_\phi(\mu(t), t) dt$
- 32: $I_{\text{pred}} \leftarrow \text{Rasterize}(\tilde{\mathbf{G}}^{t_{k+1}})$
- 33: Compute $L \leftarrow \text{loss}(I_{\text{pred}}, I^{t_{k+1}})$
- 34: Update F_ϕ
- 35: **end for**
- 36: **Output:** Optimized F_ϕ

References

- [1] Simeon Adebola, Shuangyu Xie, Chung Min Kim, Justin Kerr, Bart M van Marrewijk, Mieke van Vlaardingen, Tim van Daalen, EN van Loo, Jose Luis Susa Rincon, Eugen Solowjow, et al. Growssplat: Constructing temporal digital twins of plants with gaussian splats. *arXiv preprint arXiv:2505.10923*, 2025. 2, 3
- [2] John Charles Butcher. A history of Runge-Kutta methods. *Applied numerical mathematics*, 20(3):247–260, 1996. 4
- [3] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2, 4, 1
- [4] Nived Chebrolu, Thomas Läbe, and Cyrill Stachniss. Spatio-temporal non-rigid registration of 3d point clouds of plants. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3112–3118. IEEE, 2020. 3
- [5] Ricky TQ Chen. torchdiffeq. 2018. URL <https://github.com/rtqichen/torchdiffeq>, 14, 2018. 2, 1
- [6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Adv. Neural Inform. Process. Syst.*, 31, 2018. 2, 5
- [7] Enrico Coen and Daniel J Cosgrove. The mechanics of plant morphogenesis. *Science*, 379(6631):eade8055, 2023. 2
- [8] Stijn Dhondt, Nathalie Wuyts, and Dirk Inzé. Cell to whole-plant phenotyping: the best is yet to come. *Trends in plant science*, 18(8):428–439, 2013. 1
- [9] Jing Dong, John Gary Burnham, Byron Boots, Glen Rains, and Frank Dellaert. 4d crop monitoring: Spatio-temporal reconstruction for agriculture. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3878–3885. IEEE, 2017. 3
- [10] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021. 3
- [11] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2
- [12] Bardienus P Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Jenny Seidenschwarz, Mike Zheng Shou, Deva Ramanan, Shuran Song, Stan Birchfield, Bowen Wen, et al. Deformgms: Scene flow in highly deformable scenes for deformable object manipulation. *arXiv preprint arXiv:2312.00583*, 2023. 2
- [13] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in neural information processing systems*, 32, 2019. 3
- [14] Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ODE: the world of Jacobian and kinetic regularization. In *International conference on machine learning*, pages 3154–3164. PMLR, 2020. 3
- [15] 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*, pages 12479–12488, 2023. 2
- [16] Pawan Goyal and Peter Benner. Neural odes with irregular and noisy data. *arXiv preprint arXiv:2205.09479*, 2022. 3
- [17] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024. 2
- [18] Jacob Kelly, Jesse Bettencourt, Matthew J Johnson, and David K Duvenaud. Learning differential equations that are easy to solve. *Advances in Neural Information Processing Systems*, 33:4370–4380, 2020. 3
- [19] 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. 2, 3, 4
- [20] Patrick Kidger, Ricky TQ Chen, and Terry J Lyons. "hey, that's not an ode": Faster ode adjoints via seminorms. In *ICML*, pages 5443–5452, 2021. 3
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [22] Martin Wilhelm Kutta. Beitrag zur näherungsweisen integration totaler differentialgleichungen. *Zeitschrift für Mathematik und Physik*, 46:435–453, 1901. 4
- [23] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024. 2
- [24] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024. 2
- [25] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from dynamic scenes. *ICLR*, 5:6, 2025. 2
- [26] Luca Lobefaro, Meher VR Malladi, Tiziano Guadagnino, and Cyrill Stachniss. Spatio-temporal consistent mapping of growing plants for agricultural robots in the wild. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6375–6382. IEEE, 2024. 3
- [27] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 2, 3, 5, 7
- [28] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2

- 563 [29] Alexander Norcliffe and Marc Peter Deisenroth. Faster training
564 of neural odes using gau {\ss}-legendre quadrature. *arXiv preprint arXiv:2308.10644*, 2023. 3
- 565 [30] Haolin Pan, Franck Hétroy-Wheeler, Julie Charlaix, and
566 David Colliaux. Multi-scale space-time registration of growing
567 plants. In *2021 International Conference on 3D Vision (3DV)*, pages 310–319. IEEE, 2021. 3
- 568 [31] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien
569 Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo
570 Martin-Brualla. Nerfies: Deformable neural radiance fields.
571 In *Proceedings of the IEEE/CVF international conference on
572 computer vision*, pages 5865–5874, 2021. 2
- 573 [32] Michael P Pound, Jonathan A Atkinson, Alexandra J
574 Townsend, Michael H Wilson, Marcus Griffiths, Aaron S
575 Jackson, Adrian Bulat, Georgios Tzimiropoulos, Darren M
576 Wells, Erik H Murchie, et al. Deep machine learning pro-
577 vides state-of-the-art performance in image-based plant phe-
578 notyping. *Gigascience*, 6(10):gix083, 2017. 1
- 579 [33] Manuel García Rincón, Diego Mendez, and Julian D Col-
580 orado. Four-dimensional plant phenotyping model integrat-
581 ing low-density lidar data and multispectral images. *Remote
582 Sensing*, 14(2):356, 2022. 1
- 583 [34] Yulia Rubanova, Ricky TQ Chen, and David K Duve-
584 naud. Latent ordinary differential equations for irregularly-
585 sampled time series. *Advances in neural information pro-
586 cessing systems*, 32, 2019. 3
- 587 [35] Carl Runge. Über die numerische auflösung von differential-
588 gleichungen. *Mathematische Annalen*, 46:167–178, 1895. 4
- 589 [36] Johannes L Schonberger and Jan-Michael Frahm. Structure-
590 from-motion revisited. In *Proceedings of the IEEE con-
591 ference on computer vision and pattern recognition*, pages
592 4104–4113, 2016. 1
- 593 [37] Edmund Ware Sinnott. Plant morphogenesis. 1960. 2
- 594 [38] Eben Upton and Gareth Halfacree. *Raspberry Pi user guide*.
595 John Wiley & Sons, 2016. 1
- 596 [39] Daniel Wang, Patrick Rim, Tian Tian, Alex Wong, and
597 Ganesh Sundaramoorthi. Ode-gs: Latent odes for dy-
598 namic scene extrapolation with 3d gaussian splatting. *arXiv
599 preprint arXiv:2506.05480*, 2025. 3
- 600 [40] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi
601 Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruc-
602 tion from a single video. *arXiv preprint arXiv:2407.13764*,
603 2024. 2
- 604 [41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Si-
605 moncelli. Image quality assessment: from error visibility to
606 structural similarity. *IEEE transactions on image processing*,
607 13(4):600–612, 2004. 1
- 608 [42] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng
609 Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang.
610 4d gaussian splatting for real-time dynamic scene rendering.
611 In *Proceedings of the IEEE/CVF conference on computer vi-
612 sion and pattern recognition*, pages 20310–20320, 2024. 2,
613 3, 4, 5, 1
- 614 [43] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing
615 Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-
616 fidelity monocular dynamic scene reconstruction. In *Pro-
617 ceedings of the IEEE/CVF conference on computer vision
618 and pattern recognition*, pages 20331–20341, 2024. 2
- 619 [44] Zeyu Yang, Zijie Pan, Xiatian Zhu, Li Zhang, Yu-Gang
620 Jiang, and Philip HS Torr. 4d gaussian splatting: Model-
621 ing dynamic scenes with native 4d primitives. *arXiv preprint
622 arXiv:2412.20720*, 2024. 2, 5, 7
- 623 [45] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen,
624 Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey
625 Hu, Matthew Tancik, et al. gsplat: An open-source library for
626 gaussian splatting. *Journal of Machine Learning Research*,
627 26(34):1–17, 2025. 1
- 628 [46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shecht-
629 man, and Oliver Wang. The unreasonable effectiveness of
630 deep features as a perceptual metric. In *Proceedings of the
631 IEEE conference on computer vision and pattern recogni-
632 tion*, pages 586–595, 2018. 1
- 633 [47] Qian Zheng, Xiaochen Fan, Minglun Gong, Andrei Sharf,
634 Oliver Deussen, and Hui Huang. 4d reconstruction of bloom-
635 ing flowers. In *Computer Graphics Forum*, pages 405–417.
636 Wiley Online Library, 2017. 3
- 637 [48] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and
638 Markus Gross. Ewa volume splatting. In *Proceedings Visu-
639 alization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 3
- 640 [641]