Supplementary Material for MotionScale: Reconstructing Appearance, Geometry, and Motion of Dynamic Scenes with Scalable 4D Gaussian Splatting

In this supplementary material, we provide additional information to complement the main paper. First, we present the implementation details and experimental settings of MotionScale in Sec. A. Then, Sec. B provides an extended analysis of the efficiency and scalability of our method. In Sec. C, we include supplementary ablation studies with additional visualizations.

A. Implementation Details

A.1. Pre-processing

We leverage several off-the-shelf 2D models to extract priors that facilitate the initialization of our 4DGS reconstruction. For in-the-wild videos from the DAVIS dataset [5], we first estimate per-frame depth maps and camera poses using the geometry model π^3 [11]. We then refine the camera extrinsics via bundle adjustment using MegaSaM [4], which also estimates the camera intrinsics. To obtain dense long-range correspondences, we apply Cotracker3 [3], sampling points from the initial foreground masks and tracking them across all frames. The foreground masks are generated by SAM2 [6], using the first-frame mask as the segmentation prompt. For the DyCheck dataset [1], we use camera poses from COLMAP [7] and metric depth maps preprocessed by the pipeline of [9].

A.2. Three-stage Optimization

The MotionScale optimization pipeline consists of three stages, which allow the model to efficiently extend to new frames and progressively refine the motion learned from 2D priors.

Stage 1. The first step is to extend the model to new frames by aligning the existing motion field with the new visual information. We begin by selecting a short temporal window consisting only of the most recent T_{prop} frames. Older frames are intentionally excluded at this stage, since the motion has not yet been propagated to the new frames and long-range interactions could otherwise introduce instability. For each newly added frame, we initialize its motion field by copying the motion parameters from the most recent optimized frame. We then perform a lightweight optimization that updates only the motion field of the new frames while keeping all other parameters fixed, including the Gaussian attributes. This focuses the update on foreground motion and significantly reduces computation.

During this optimization, we employ the one-directional tracking loss (Eq. 5 in the main paper), where t is sampled from recent frames in the propagation window and t' from the new frames. In addition, we supervise the update using an RGB reconstruction loss and a depth consistency loss. To restrict the update to the foreground region, we incorporate a foreground mask loss, and we further enforce temporal coherence through a temporal smoothness loss.

To encourage locally rigid motion in dynamic regions, we apply an as-rigid-as-possible (ARAP) regularization over the dynamic Gaussians. We first randomly sample N dynamic Gaussian centers $\{\mu_i^t\}_{i=1}^N$ at time t. For each sampled point μ_i^t, we construct a local neighborhood $\mathcal{N}(i)$ using its k-nearest Gaussians in 3D. Since trajectories are available for all points in the window, we estimate local rigidity by comparing their velocities. For each point i, we construct a concatenated velocity vector $\mathbf{v}_i$ by stacking its frame-to-frame displacements over T_{prop} frames in the propagation window. For any neighbor $j \in \mathcal{N}(i)$, we measure the motion discrepancy:

$$d_{ij} = \frac{1}{T_{\text{prop}}} \sum_t \|\mathbf{v}_i - \mathbf{v}_j\|. \tag{1}$$

We then convert this discrepancy into a rigidity weight using a Gaussian kernel:

$$w_{ij} = \exp(-\frac{d_{ij}^2}{2\sigma_{\text{arap}}^2}), \tag{2}$$

where σ_{arap}^2 controls sensitivity to motion variation. Pairs that move similarly receive high weights, while highly deforming or non-rigid regions are naturally down-weighted.

For each patch $\mathcal{N}(i)$, we estimate the best-fitting rigid transformation that aligns their warped positions. Let μ_j^t denote the Gaussian center at a recent frame t and $\mu_j^{t'}$ its

warped position in the new frame t'. The optimal rigid transform $\mathbf{R}_i$ is obtained by solving a weighted Procrustes problem:

$$\mathbf{R}_i = \underset{\mathbf{R}\in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{j\in\mathcal{N}(i)} w_{ij} \|(\mu_i^{t'} - \mu_j^{t'}) - \mathbf{R}(\mu_i^t - \mu_j^t)\|^2. \quad (3)$$

Finally, given the estimated local rigid transform, the ARAP loss penalizes deviations from rigidity within each neighborhood:

$$\mathcal{L}_{\text{ARAP}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j\in\mathcal{N}(i)} w_{ij} \|(\mu_i^{t'} - \mu_j^{t'}) - \mathbf{R}_i(\mu_i^t - \mu_j^t)\|^2. \quad (4)$$

This encourages each local patch to undergo approximately rigid motion, while allowing non-rigid regions to remain flexible through their automatically reduced weights w_{ij}.

Stage 2. Once the motion is roughly aligned in Stage 1, we further refine the motion field using a denser and more symmetric supervision. In this stage, we apply a two-directional tracking loss over arbitrary frame pairs (t, t') within the propagation window, allowing both forward and backward correspondences to jointly constrain the local motion. The remaining losses (RGB, depth, mask, temporal smoothness, and ARAP) follow the same formulation as in Stage 1, but are now evaluated over a larger temporal neighborhood. Importantly, we also update the motion parameters of the recent frames in the window, enabling them to benefit from the information introduced by the newly incorporated frames. New frames are progressively added to the model by alternating between Stage 1 and Stage 2. After Stage 1 brings the new frames into the coarse alignment, Stage 2 refines motion for both the new frames and previously added ones within the propagation window. This iterative process incrementally propagates and stabilizes motion across the sequence. After a sufficient number of frames have been incorporated and locally refined, we proceed to Stage 3 for long-term motion optimization.

Stage 3. After a sufficient number of frames have been incorporated through Stages 1 and 2, we perform a global optimization over all frames in the sequence. In contrast to the first two stages, Stage 3 enables full Gaussian densification. We delay the densification until this stage to avoid destabilizing the motion field, since Stages 1 and 2 supervise only a limited temporal range, and adding new Gaussians can disrupt the aligned motion in earlier frames. We also introduce a normal consistency loss to improve geometric accuracy. We estimate Gaussian surface normals by computing per-pixel normals from the rendered depth map, and supervise these using normal maps predicted by MoGe [10]. This normal loss encourages smoother and more coherent geometry, particularly in regions where depth gradients are ambiguous. Together with the RGB photometric loss, the ARAP regularization, and all the auxiliary terms defined, Stage 3 performs a global refinement over the entire sequence that finalizes both the geometry and motion, completing the MotionScale optimization pipeline.

Table 1. Comparison of training and rendering speed on the DyCheck [1] dataset. "Time" denotes the average time required to process 100 frames on each dataset, measured in minutes, while "FPS"" represents the average frame rate during test rendering.

Method	Time↓	FPS↑	PSNR↑	SSIM↑	LPIPS↓
4D Gaussians [12]	22.1	76	13.11	0.39	0.73
Dynamic GM [8]	187.6	172	15.79	0.59	0.44
Shape of Motion [9]	109.4	39.2	16.72	0.63	0.45
MotionScale (Ours)	115.6	43.6	17.98	0.70	0.40

B. Efficiency

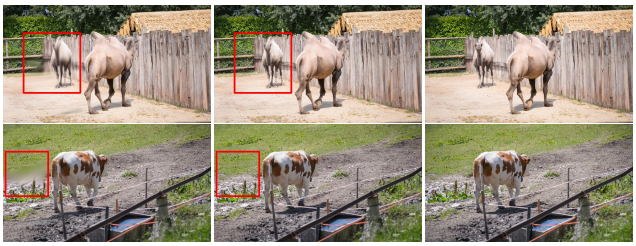
MotionScale is designed to balance reconstruction accuracy with practical training and rendering efficiency. Here, we compare the model efficiency on the DyCheck dataset by computing the optimization time and rendering FPS. As shown in Tab. 1, our total optimization time is competitive with recent dynamic 4D reconstruction methods, and our rendering speed remains real-time. While our training time is slower than 4D Gaussians due to the additional motion modeling, it is substantially faster than Dynamic GM [8] and comparable to Shape of Motion [9], while achieving significantly higher reconstruction quality across all metrics. Our method achieves this efficiency through the staged optimization strategy. By restricting Stages 1 and 2 to a short propagation window and updating only the motion parameters, we avoid full-sequence optimization until Stage 3. This greatly reduces computation in early iterations and ensures that newly added frames can be assimilated efficiently. In addition, our scalable motion field allows MotionScale to handle long video sequences without a proportional increase in computational cost.

C. Ablation Studies

Adaptive Control. In the main paper, we evaluated the quantitative impact of the adaptive control mechanism on our cluster-based motion bases. Here, we provide additional visual comparisons to illustrate its effect. This mechanism allows the motion bases to gradually grow and redistribute across the entire object, enabling them to capture fine-grained deformation as new frames are incorporated. Without it, the motion bases remain largely fixed to their initialization and fail to expand to regions with complex motion. This leads to an imbalance between clusters and spatial coverage, ultimately degrading motion reconstruction quality. Fig. 1 shows side-by-side visualizations of the learned clusters and the rendered RGB images, comparing

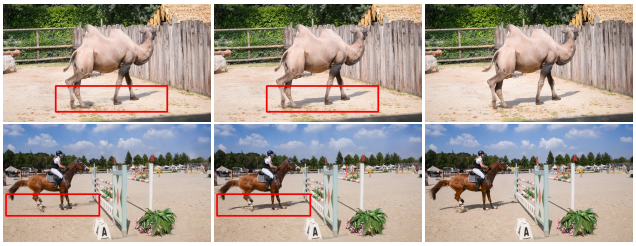
(a) GT (b) Clusters (Frame10) (c) RGB (Frame10) (d) Clusters (Frame50) (e) RGB (Frame50)

Figure 1. Visual comparison of adaptive control. We compare our full pipeline (top row) with an ablation variant without adaptive control (bottom row). We show that in the early stages (b-c), our adaptive mechanism can effectively split non-rigid clusters to capture fine-grained motion. In later stages (d-e), it gradually expands to previously unseen surfaces, allowing the motion bases to cover new geometry as it becomes visible.



(a) w/o Sampling (b) w/ Sampling (c) GT

Figure 2. Visual comparison of background propagation. (a) Without sampling, newly revealed background regions appear as holes. (b) With sampling, these areas are correctly reconstructed. (c) Ground truth image.



(a) w/o Shadows (b) w/ Shadows (c) GT

Figure 3. Shadow Gaussian ablation. Removing shadow Gaussians (a) produces ambiguous or missing shadows, while our model (b) reconstructs clear, stable shadows close to the ground truth (c).

MotionScale with and without the adaptive control process. Without adaptivity, clusters collapse or remain static, and the reconstructed motion exhibits distortions and temporal inconsistencies. In contrast, our adaptive bases grow to cover the full object, resulting in significantly better alignment and sharper dynamic details.

Background Propagation. We also include visual evidence for the effectiveness of our background propagation strategy. As described in the main paper, newly revealed regions in the background (e.g., areas previously occluded by moving foreground objects) must be sampled and initialized properly in order to produce a complete scene reconstruction. If background propagation is disabled, the Gaussian field cannot recover newly visible background surfaces, leading to noticeable holes or empty regions behind moving objects. These artifacts become especially apparent when the foreground undergoes large displacements or reveals new parts of the scene. Fig. 2 presents comparisons of the rendered RGB images with and without background propagation. Without this mechanism, substantial gaps appear in the background and remain unfilled across subsequent frames. With background propagation enabled, MotionScale successfully populates newly visible regions, ensuring a complete and coherent reconstruction.

Shadow Gaussians. We further examine the impact of our shadow Gaussian modeling. Standard 4DGS [2, 9] treats shadows as part of the background, which prevents accurate reconstruction when shadows move with the foreground object. In contrast, our shadow Gaussians explicitly model cast shadows as a separate component, resulting in much cleaner shading and reducing ambiguity near the object boundary. Visual comparisons in Fig. 3 show that removing shadow Gaussians leads to smeared or ambiguous shadow regions, and in many cases the cast shadow is partially missing. Our full model can recover clear and stable shadows, which in turn supports more accurate dynamic reconstruction.

References

[1] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022. 1, 2

[2] 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. 3

[3] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025. 1

[4] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10486–10496, 2025. 1

[5] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 1

[6] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[7] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[8] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2

[9] Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9660–9672, 2025. 1, 2, 3

[10] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5261–5271, 2025. 2

[11] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning, 2025. 1

[12] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024. 2