Supplementary Material for *Catalyst4D: High-Fidelity 3D-to-4D Scene Editing via Dynamic Propagation*

A. Video Demo

We provide demo videos on our project page showing the dynamic editing capabilities of our method. The videos include comparative results against baseline approaches on novel test viewpoints, as well as demonstrations of image-guided global style transfer and text-guided local edits enabled by the proposed Catalyst4D. We encourage readers to view these videos, as they offer a clear illustration of the high visual fidelity and editing consistency achieved by our approach.

B. Additional Comparison Results

B.1. Additional Quantitative Comparison

Standard CLIP and VBench scores may fail to capture fine-grained 3D-to-4D propagation due to their global nature, we further adopt **EditScore** (instruction-following accuracy and visual fidelity) [6] and **VE-Bench** (local temporal consistency) [8], better aligned with human perception. Tab. B.1 on **16** prompts(multi-camera) from the main paper shows Catalyst4D consistently surpasses prior methods.

B.2. Additional Quantitative Ablation

We clarify that the main paper ablation (Tab. 2) is cumulative: w/o AMG is the "w/o both" baseline, and w/o CUAR retains only AMG. Tab. B.2, evaluated with new metrics, further confirms both AMG and CUAR are essential, outperforming DeformNet-Guide (w/o both).

B.3. Comparison on Background Preservation

Although the compared CTRL-D [3] appears visually plausible, it still introduces undesired modifications in non-edited regions due to the limitations of the underlying 2D diffusion model. As illustrated in Fig.B.1, while CTRL-D edits the character, it also causes noticeable deviations in non-target objects such as the dog on the stool and objects on the table compared to the original scene. By contrast, our method constrains target dynamic Gaussians directly through 3D editing gradients, enabling more precise and localized editing in dynamic scenes.

B.4. Comparison on the Monocular Dataset

We further evaluate text-driven 4D editing on the monocular HyperNeRF [7] dataset. Since Instruct 4D-to-4D does not support this dataset, we primarily compare our method against Instruct-4DGS and CTRL-D. The qualitative comparisons are shown in Fig. B.2, where the top and bottom rows of each example correspond to different timesteps. Instruct-4DGS struggles to localize the target object, resulting in edits being incorrectly applied to irrelevant regions of the scene. CTRL-D, which reconstructs the 4D representation from 2D diffusion outputs, suffers from the inherent modality gap, producing blurry results and noticeable temporal inconsistency. As illustrated in the "Glacial extruder" example in the lower half of Fig. B.2, the appearance of the edited extruder varies substantially across timesteps. In contrast, our method directly propagates a high-fidelity 3D edit from the first frame to all subsequent frames, yielding clearer textures, stronger temporal coherence, and significantly improved overall 4D editing quality.

Table B.1. Quantitative comparison using EditScore and VE-Bench.

Metric	IN4D	I4DGS	CTRL-D	Ours
EditScore↑	4.034	5.618	4.326	**7.375**
VE-Bench↑	0.155	0.256	0.163	**1.080**

Table B.2. Ablation studies on AMG and CUAR modules.

Metric	w/o both	w/ AMG	w/ CUAR	w/ both
EditScore↑	2.713	5.112	5.005	**7.375**
VE-Bench↑	0.059	0.452	0.580	**1.080**

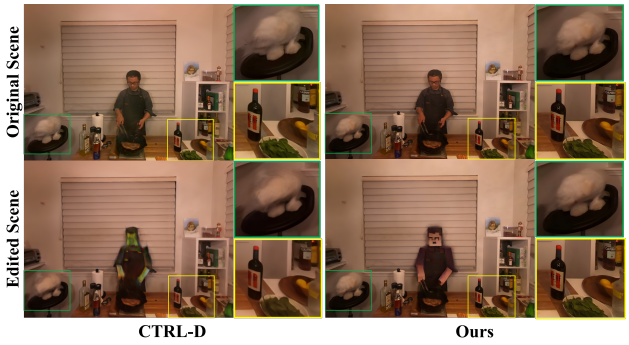

Figure B.1. Qualitative comparison of localized editing. In contrast to CTRL-D, which introduces inconsistencies in non-edited regions, our method achieves more precise and localized editing by constraining dynamic Gaussians via 3D editing gradients.

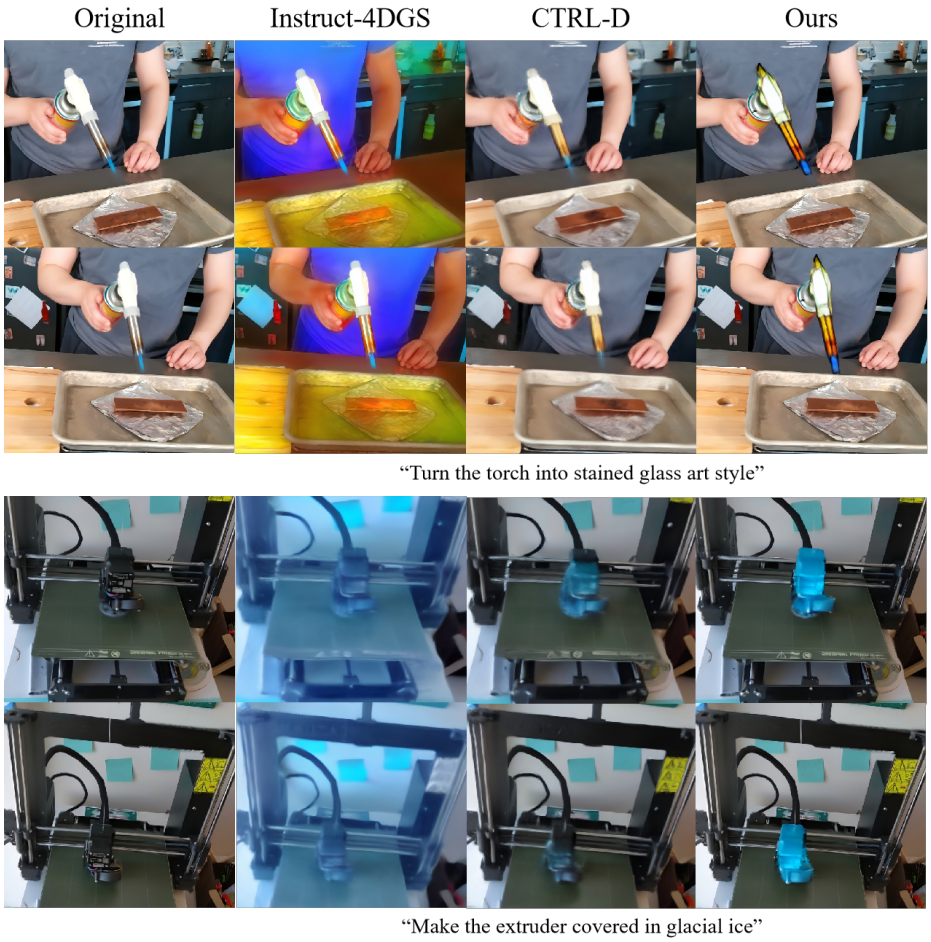

Figure B.2. Comparison on HyperNeRF Dataset. The results show that Instruct-4DGS fails to correctly localize edits and CTRL-D produces blurry and temporally inconsistent results, whereas our method demonstrates significant advantages in both visual quality and temporal consistency.

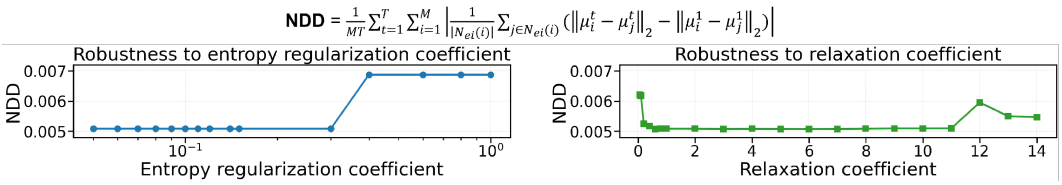

Figure B.3. Effect of Sinkhorn regularization on motion transfer.

B.5. Robustness of Sinkhorn Algorithm

Our outlier-filtered anchors yield a well-conditioned cost matrix, enabling stable Sinkhorn convergence. To assess robustness under extreme topology changes (e.g. *Searsteak* to *Minecraft*), we measure Neighborhood Distance Deviation (NDD), which captures the temporal preservation of local anchor geometry and directly reflects correspondence stability. Fig. B.3 confirms NDD stability across wide Sinkhorn ranges ($\lambda_0 \in [0.05, 0.3], \lambda_{1,2} \in [0.6, 11]$).

B.6. Comparison of Global Style Transfer

We further compare our method with CLIPGaussian [4] and CTRL-D on the task of image-guided global style transfer. To establish a controlled comparison with CTRL-D, which requires an image prompt, we first stylize the first frame 3D scene using SGSST [2], our 3D baseline. A rendered image from this stylized 3D scene is then provided to CTRL-D as its style reference. Consequently, our method and CTRL-D begin with nearly identical style information but leverage

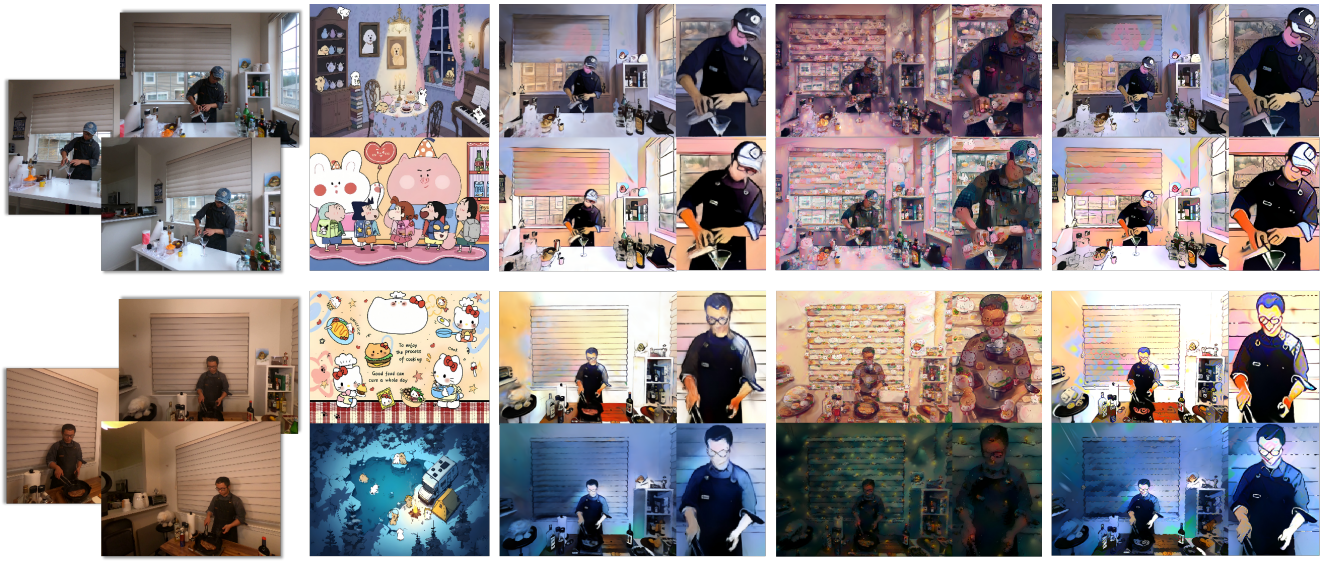

Figure B.4. Comparative results for global style transfer. Unlike the over-smoothed results of CTRL-D and the chaotic textures from CLIP-Gaussian, our method produces finer visual textures, better preserves the original scene's geometry, and demonstrates a color distribution more consistent with the reference style image.

distinct technical approaches. Our method propagates the style from the 3D scene across the sequence (a 3D-to-4D approach), while CTRL-D uses its DreamBooth-based fine-tuning to reconstruct the 4D effect from the 2D style image (a 2D-to-4D approach). This allows for a direct assessment of the two pathways, minimizing the influence of external stylization models.

As shown in Fig. B.4, the global style transfer results produced by CTRL-D appear superficially similar to ours, an expected outcome given that CTRL-D's style reference is a rendered image from our 3D stylization baseline. However, our method achieves noticeably finer textures and higher visual clarity. CTRL-D, which relies on a 2D diffusion model, often generates results with blurriness and over-smoothing due to the inherent 2D-to-4D reconstruction gap. Meanwhile, CLIPGaussian tends to introduce chaotic textures, as reflected in the distorted facial region of the character in the example shown in the fourth column of Fig. B.4. In contrast, our approach preserves the underlying 4D geometry more effectively by propagating SGSST [2]-based stylization (our 3D baseline) consistently across the sequence. Additionally, our results exhibit a color distribution that more faithfully matches the reference style image.

C. Implementation Details

C.1. Optimal Transport-based Anchor Matching

Given the anchor set $A_{\text{src}} = \{\mathbf{p}_i^{\text{src}}\}_{i=1}^{n}$ of the original dynamic Gaussians $\mathcal{G}^1$ in the first frame and the anchor set $A_{\text{edit}} = \{\mathbf{p}_j^{\text{edit}}\}_{j=1}^{m}$ of the edited Gaussians $\mathcal{G}^1_{\text{edit}}$, we formulate the anchor correspondence as an unbalanced optimal transport (UOT) problem and solve it using the Sinkhorn algorithm [1].

We define a distance matrix D using a Welsch robust penalty:

$$D_{ij} = 1 - \exp\left(-\frac{\|\mathbf{p}_i^{\text{src}} - \mathbf{p}_j^{\text{edit}}\|_2^2}{2\beta^2}\right), \tag{1}$$

where $\beta = \gamma \cdot d_{\text{med}}$, d_{med} is the median of all pairwise anchor distances, and we empirically set $\gamma = 0.05$.

The UOT objective is written as:

$$\min_{P \in \mathbb{R}_+^{n\times m}} \sum_{i=1}^{n}\sum_{j=1}^{m} D_{ij}\mathbf{P}_{ij} - \lambda_0 \sum_{i,j} \mathbf{P}_{ij}\log \mathbf{P}_{ij} + \lambda_1 \,\mathrm{KL}\left(\mathbf{P}\mathbf{1}_m \,\middle\|\, \frac{1}{n}\mathbf{1}_n\right) + \lambda_2 \,\mathrm{KL}\left(\mathbf{P}^{\mathrm{T}}\mathbf{1}_n \,\middle\|\, \frac{1}{m}\mathbf{1}_m\right), \tag{2}$$

s.t. $\mathbf{P}_{ij} \geq 0, \forall i \in \{1,...,n\}, j \in \{1,...,m\}$, where $\mathbf{P}$ is the transport plan matrix, $\lambda_0 = 0.1$, $\lambda_1 = 1.0$, $\lambda_2 = 1.0$ are regularization weights, and KL denotes the Kullback-Leibler divergence.

From the optimal plan $\mathbf{P}^*$, we obtain the correspondence $corr$ by assigning each edited anchor to the source anchor with the highest transport mass:

$$corr_j = \arg\max_{i\in\{1,...,n\}} \mathbf{P}^*_{ij}. \tag{3}$$

C.2. Deformation of Edited Dynamic Gaussians

Notably, when computing the deformation of edited Gaussians, unlike the additive form used for positional deformation $\Delta\mu$ and rotational deformation $\Delta\mathbf{q}$, scaling deformation $\Delta\mathbf{s}$ is defined multiplicatively to preserve proportional change:

$$\begin{aligned} \mu^t &= \mu^1 + \Delta\mu^t, \\ \mathbf{q}^t &= \mathbf{q}^1 + \Delta\mathbf{q}^t, \\ \mathbf{s}^t &= \mathbf{s}^1 \cdot \Delta\mathbf{s}^t. \end{aligned} \tag{4}$$

where $\Delta\mu^t$, $\Delta\mathbf{q}^t$, $\Delta\mathbf{s}^t$ are the deformation quantities of edited Gaussians from frame 1 to frame t.

C.3. Additional Experimental Details

Anchor Construction. Since $\mathcal{G}^1_{\text{edit}}$ and $\mathcal{G}^1$ are largely spatially aligned, we compute a shared bounding sphere and sample an identical set of rays to construct anchor points for both Gaussian clouds. Specifically, we sample 300,000 rays from the minimum bounding sphere and use $k = 2$ nearest neighbors when forming local anchor neighborhoods. For Gaussians not covered by any anchor region, we assign them to the nearest anchor using Euclidean distance.

Loss Design. For foreground edits, we use standard L1+SSIM to propagate the effect faithfully. Background supervision relies on L1 only, as CUAR freezes non-edited Gaussians and L1 sufficiently prevents artifacts; adding SSIM yields negligible gains while increasing cost.

Appearance Refinement Parameters. Eq. 17 exploits uncertainty contrast between artifact-prone and stable regions. The threshold ϵ trades quality for efficiency (higher for faster refinement, lower for broader coverage) and remains stable over a reasonable range. To flexibly control the spatial coverage of the artifact mask, we set the threshold parameter ϵ within the range $[0.5, 2.0]$. During refinement, the loss weights are fixed as $\eta = 0.2$ and $\zeta = 0.3$, applied to the foreground and background objectives described in Sec. 4.2. These settings balance artifact correction with preservation of non-corrupted regions.

Evaluation Protocol and Implementation. For quantitative evaluation, we render novel test viewpoints and compute the CLIP similarity by averaging scores over 30 uniformly sampled frames from each test video. Temporal consistency is evaluated on the full rendered sequence to measure stability across viewpoints and timesteps.

For dynamic scene reconstruction with Swift4D, we adopt effective rank regularization [5] to suppress needle-like artifacts. Following the settings of 4DGS [9], we constrain the edited dynamic Gaussians to share consistent opacity and color attributes across all frames during optimization.

The training breakdown is as follows: Anchor construction (<30s), Sinkhorn solver (∼15s), Motion Guidance (∼1 min), and CUAR (25–35 min). All experiments are implemented in Python 3.7.12 using PyTorch 1.13.1 on Ubuntu 22.04. Our method is trained on a single NVIDIA A100 Tensor Core GPU.

C.4. Editing Prompts and 3D Baseline Configurations

Here, we provide the detailed editing configurations used in our experiments, including the text prompts and the corresponding 3D editing baselines adopted for each edit.

For the results shown in Fig. 1, the editing instructions and 3D baseline methods are:

- "Turn the torch into Pop art style" with DreamCatalyst,
- "Make the torch carved from a flawless emerald" with DGE,
- "Turn him into a Claymation character" with DGE,
- "Make the extruder covered in glacial ice" with DreamCatalyst,
- "Make the extruder look like toast" with DGE,
- "Turn him into JoJo's Bizarre Adventure anime style" with DGE.

For the edits in Fig. 4, we use:

- "Turn his clothes into a football player outfit" with DreamCatalyst,
- "Turn his hat into a newsboy cap" with DGE,
- "Turn him into a Minecraft character" with DGE,
- "Make them look like Marble roman sculptures" with DreamCatalyst,
- "Turn him into Deadpool" with DGE.

For the results of our method reported in Tab. 1, all 3D edits are performed using DGE. The applied prompts are:

- **Sear-steak.** "Turn him into a Minecraft character", "Turn him into Super Mario",
- **Coffee-martini.** "Turn his hat into a newsboy cap", "Turn him into JoJo's Bizarre Adventure anime style",
- **Trimming.** "Change the leaves to autumn leaves", "Turn him into Deadpool".

For the results in Tab. 2, the editing setups are:

- "Turn him into a Minecraft character" with DGE,
- "Turn him into Batman" with DreamCatalyst,
- "Turn him into JoJo's Bizarre Adventure anime style" with DGE.

D. Limitation

While Catalyst4D effectively extends high-quality 3D editing capabilities to dynamic 4D scenes, it inherits certain dependencies from the underlying components or pretrained models.

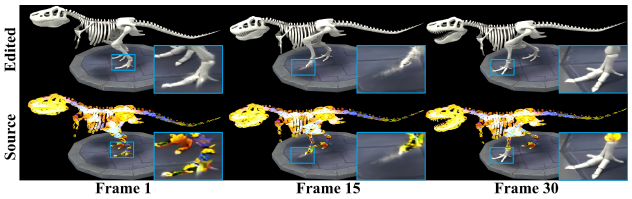

Figure D.1. Failure case (motion jitter illustration).

First, the temporal coherence of our method is influenced by the quality of the initial 3D edits. If the 3D editing results lack sufficient spatial consistency, this may propagate and affect the spatio-temporal consistency in the final dynamic outputs. Moreover, as our method does not modify the deformation network or re-optimize Gaussian densities, it relies on the stability of the underlying 4D reconstruction. Under severe reconstruction noise (e.g., point jitter or low-opacity Gaussians), motion guidance may be locally disrupted. In Fig. D.1 (*D-NeRF, trex*), background Gaussians drift into the edited foreground, causing local artifacts, reflecting a shared limitation of current 4D reconstruction models. Nonetheless, such limitations are not fundamental and can be mitigated as upstream 3D editing and 4D reconstruction techniques continue to advance. Our framework is fully compatible with future improvements in both areas.

References

[1] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 3

[2] Bruno Galerne, Jianling Wang, Lara Raad, and Jean-Michel Morel. Sgsst: Scaling gaussian splatting style transfer. In *CVPR*, pages 26535–26544, 2025. 2, 3

[3] Kai He, Chin-Hsuan Wu, and Igor Gilitschenski. Ctrl-d: Controllable dynamic 3d scene editing with personalized 2d diffusion. In *CVPR*, pages 26630–26640, 2025. 1

[4] Kornel Howil, Piotr Borycki, Tadeusz Dziarmaga, Marcin Mazur, PrzemysĹ Spurek, et al. Clipgaussian: Universal and multimodal style transfer based on gaussian splatting. *arXiv preprint arXiv:2505.22854*, 2025. 2

[5] Junha Hyung, Susung Hong, Sungwon Hwang, Jaeseong Lee, Jaegul Choo, and Jin-Hwa Kim. Effective rank analysis and regularization for enhanced 3d gaussian splatting. In *NeurIPS*, pages 110412–110435, 2024. 4

[6] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, et al. Editscore: Unlocking online rl for image editing via high-fidelity reward modeling. *arXiv preprint arXiv:2509.23909*, 2025. 1

[7] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 1

[8] Shangkun Sun, Xiaoyu Liang, Songlin Fan, Wenxu Gao, and Wei Gao. Ve-bench: Subjective-aligned benchmark suite for text-driven video editing quality assessment. In *AAAI*, pages 7105–7113, 2025. 1

[9] 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 *CVPR*, pages 20310–20320, 2024. 4