Tempo-SAM3D: Monocular Video to 4D via Temporal Memory-Guided Generation

Baicheng Li[1], Dong Wu[1], Yingdian Cao[1], Haoxiang Yang[1], Yiwen Lu[1], Zike Yan[2], and Hongbin Zha[1,3⋆]

[1] State Key Laboratory of GAI, School of IST, Peking University, Beijing, China
[2] T-Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong SAR, China
[3] School of Artificial Intelligence and Computer Science, Anqing Normal University, Anqing, China

Abstract. Generating temporally coherent 4D content from monocular video is a challenging yet practically important problem. Recent advances in layout-aware 3D generation have enabled producing objects with their spatial arrangements from a single image, naturally providing scene-level understanding that is valuable for 4D generation. However, applying such single-image models independently to each video frame yields severe temporal inconsistencies—flickering textures, structural discontinuities, and jittering poses. We present **Tempo-SAM3D**, a training-free framework that extends layout-aware 3D generation to coherent 4D generation from monocular video. Our approach introduces Dual-Path Temporal Attention, which maintains importance-driven KV caches in both cross-attention and self-attention to propagate visual memory and structural memory across frames, respectively. We further incorporate gradient-guided velocity correction that constrains the flow matching sampling trajectory, and HexPlane-based temporal smoothing that regularizes latent representations via low-rank spatiotemporal decomposition. Tempo-SAM3D requires no model retraining and inherits layout awareness for spatially-grounded 4D generation. Experiments demonstrate clear improvements over per-frame baselines and existing video-to-4D methods.

1 Introduction

The generation of 3D content from images has advanced dramatically in recent years. Feed-forward 3D generation systems [1, 32, 37, 40, 42] can now produce high-fidelity textured 3D assets from a single image in seconds, transforming what was once a labor-intensive modeling process into an automated pipeline. A particularly notable development is the emergence of *layout-aware* 3D generation [4], which goes beyond object-centric reconstruction to recover not only the geometry and appearance of individual objects, but also their spatial arrangement within the scene—the scale, rotation, and translation that specify *where*

⋆ Corresponding author: zha@cis.pku.edu.cn

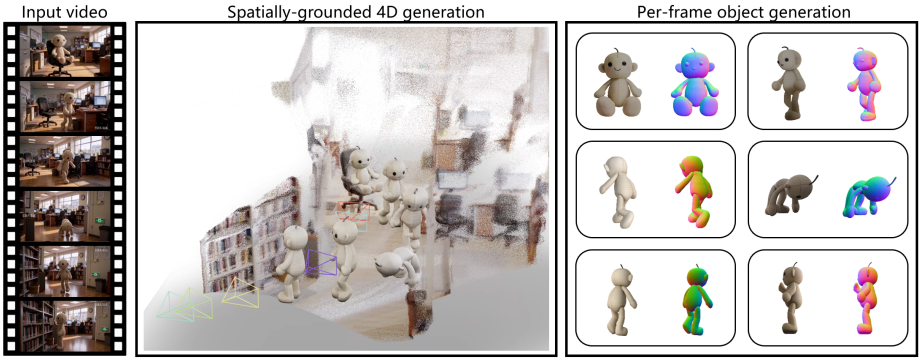

Fig. 1: Tempo-SAM3D takes a monocular video as input (left) and generates spatially-grounded 4D content: temporally consistent 3D objects placed at their correct positions within the observed scene (center), each with high-quality geometry and appearance (right).

each object resides. This scene-level understanding bridges the gap between isolated asset creation and practical scene composition.

Such layout awareness becomes especially valuable when moving from single images to video. In applications such as autonomous driving, robotics, AR/VR, and video editing, observations naturally arrive as temporal sequences. Generating dynamic 3D content from these sequences demands not only per-frame quality but also cross-frame coherence: objects must maintain consistent geometry, texture, and scene placement over time. Layout-aware generation provides a strong starting point, as it already recovers per-frame spatial information that most methods discard; yet bridging the gap from independent per-frame outputs to a coherent 4D sequence remains an open problem.

The core difficulty lies in temporal consistency. Since each frame is processed independently, the model produces different hallucinations every time—the unobserved portions of each object are imagined anew at each step. This inconsistency pervades multiple levels of the generation process: internally, the 3D latent representations lack continuity, causing structural discontinuities in unobserved regions; externally, the generated appearance and geometry drift across frames, producing texture flickering and geometric artifacts; and at the scene level, independently estimated spatial arrangements accumulate errors, leading to positional drift. Addressing temporal consistency therefore requires a solution that operates across these levels rather than at any single point.

Most prior 4D methods sidestep this issue through a *canonical-plus-deformation* paradigm: they reconstruct a shared canonical 3D model and learn a deformation field to animate it. While this yields inherently smooth results, it usually assumes fixed topology and cannot handle scenarios with large shape variations; moreover, these methods typically produce objects in isolation, without scene-level spatial context.

We take a different path. Our key insight is that temporal consistency can be injected at multiple complementary levels of a per-frame generation pipeline,

without modifying the model's weights. At the attention level, sharing KV pairs across frames propagates structural and visual memory. At the solver level, gradient-based guidance constrains trajectory smoothness. At the post-processing level, low-rank spatiotemporal decomposition filters residual noise.

Based on this insight, we present **Tempo-SAM3D**, a training-free framework that extends layout-aware 3D generation to the temporal domain. As illustrated in Fig. 1, given a monocular video, our method produces spatially-grounded 4D outputs that are geometrically detailed, temporally coherent, and precisely positioned within the observed scene. Our contributions are as follows:

- We propose Tempo-SAM3D, the first framework that lifts layout-aware single-image 3D generation to spatially-grounded 4D generation from monocular video, introducing multi-level temporal consistency mechanisms that require no model retraining.
- We introduce Dual-Path Temporal Attention, maintaining separate importance-driven KV caches in cross-attention and self-attention to propagate 2D visual memory and 3D structural memory across frames, enabling appearance preservation under viewpoint changes and structural coherence over time.
- We further incorporate gradient-guided velocity correction and HexPlane-based temporal smoothing, forming a comprehensive consistency system that jointly operates at the attention, solver, and post-processing levels.

2 Related Work

Single-Image 3D Generation. Feed-forward generation approaches, including LRM [9], InstantMesh [42], CRM [37], TripoSR [32], SF3D [1], TRELLIS [40], and SPAR3D [11], directly regress 3D representations from images in a single forward pass, achieving real-time or near-real-time performance. More recently, native 3D diffusion models such as Hunyuan3D 2.0 [52], Direct3D [39], TripoSG [20], and CraftsMan3D [19] generate high-fidelity 3D assets directly in structured latent or mesh spaces, bypassing multi-view intermediate representations. Among these, SAM3D [4] introduces *layout-aware* generation—predicting not only object geometry and texture but also each object's spatial arrangement within the scene. Our work extends SAM3D along the *temporal* axis, lifting single-image layout-aware generation to temporally coherent 4D generation from video.

4D Reconstruction and Generation. Dynamic 3D reconstruction extends neural representations to the temporal domain: D-NeRF [27] learns a canonical NeRF with a deformation field, HexPlane [2] and K-Planes [7] factorize the 4D volume into compact 2D feature planes, and Gaussian-based methods [10,18,38,46] achieve real-time dynamic rendering. Feed-forward approaches such as L4GM [29] and Any4D [13] directly predict 4D representations from images or video, while Shape of Motion [34] and DynMF [15] factorize motion from casual monocular videos. On the generation side, 4DGen [47], DreamGaussian4D [28], Consistent4D [12], and STAG4D [49] lift images to animated 3D

assets via SDS or video diffusion priors, while 4Diffusion [50] and SV4D [41] train multi-view video diffusion models for more consistent generation, and DreamMesh4D [21] introduces Gaussian-mesh hybrid representations. V2M4 [3] combines native 3D mesh generation with mesh optimization for 4D animation, and AR4D [53] proposes autoregressive 4D generation from monocular video. Unlike the canonical-plus-deformation approaches, our per-frame generation paradigm naturally handles large shape variations or topological changes, while inheriting layout awareness for scene-level spatial understanding.

Monocular Scene Reconstruction. Classical visual SLAM systems [6,26] estimate camera trajectories and build sparse maps through bundle adjustment, while neural SLAM methods [14, 17, 18, 24, 31, 48] integrate learned representations for denser reconstruction with real-time rendering capability. On the feed-forward side, monocular depth estimation has seen remarkable progress: Depth Anything [44] and its successors [22, 45] produce high-quality depth from single images using large-scale training. MoGe [35] predicts dense point maps with intrinsic camera parameters, while DUSt3R [36], MASt3R [16], VGGT [33], and Fast3R [43] reconstruct 3D scenes from image pairs or sets through direct pointmap regression. MonST3R [51] extends this paradigm to handle dynamic scenes with motion-aware geometry estimation, and Flow3R [5] further integrates factored flow prediction for scalable visual geometry learning. In our pipeline, reconstructed point maps serve as both spatial conditioning for the 3D generation backbone and geometric alignment targets for scene placement, while estimated camera trajectories enable consistent overlay of generated objects across frames.

3 Method

We present Tempo-SAM3D, a training-free framework for 4D generation from monocular video. An overview of the framework is illustrated in Fig. 2. Our approach introduces temporal consistency at three complementary levels: attention-level memory propagation via Dual-Path Temporal Attention (Sec. 3.2), solver-level gradient guidance via Temporal and Alignment Guidance (Sec. 3.3), and post-generation temporal smoothing via HexPlane decomposition (Sec. 3.4).

3.1 Preliminaries

Our framework builds upon SAM3D [4], a layout-aware 3D generation pipeline built on the TRELLIS architecture [40]. SAM3D generates 3D assets through two sequential stages, both employing flow matching [23] to learn a velocity field $v_\theta(x_s, s, c)$ that is integrated via ODE from noise toward the data distribution. At any step s, a lookahead prediction $\hat{x}_0 = x_s - s \cdot v_s$ provides an estimate of the final clean output.

Stage 1: Sparse Structure Generation. The first stage produces a sparse voxel structure $\mathcal{V}$ that captures the object's coarse geometry together with its layout pose, comprising scale, rotation, and translation within the scene. A

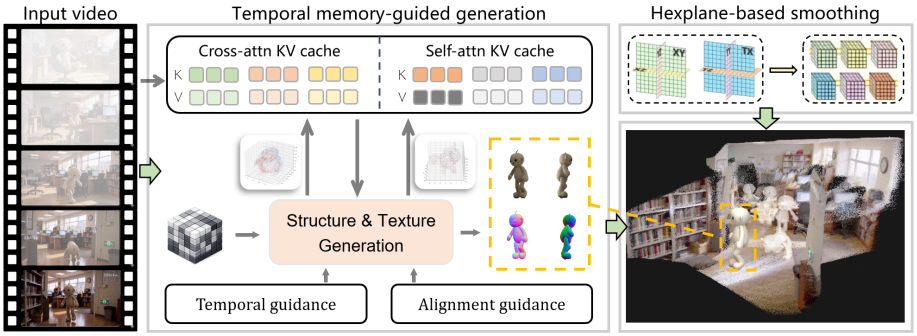

Fig. 2: Overview of Tempo-SAM3D. Each frame undergoes temporal memory-guided 3D generation. *Dual-Path Temporal KV Caches* (top) propagate structural and visual memory across frames (Sec. 3.2). *Gradient-guided velocity correction* (bottom) applies Temporal Guidance and Alignment Guidance (Sec. 3.3). *HexPlane temporal smoothing* (right) filters residual noise via low-rank spatiotemporal decomposition (Sec. 3.4).

Multi-Modal DiT jointly models shape and spatial arrangement, conditioned on visual features extracted from the input image and segmentation mask, as well as a pointmap encoding spatial correspondence between the image and the scene.

Stage 2: Structured Latent Generation. Building on the sparse structure from Stage 1, the second stage generates high-resolution structured latent representations that encode detailed geometry and texture. A DiT architecture with cross-attention to input image patch features refines the coarse voxel layout into a complete 3D asset with rich surface detail and appearance.

Crucially, SAM3D recovers not only object geometry and appearance but also the object's spatial arrangement within the scene—a capability we term *layout awareness*. We extend SAM3D along the temporal axis to enable coherent 4D generation from video, while inheriting and enhancing this layout awareness to achieve spatially-grounded generation.

3.2 Dual-Path Temporal Attention

The DiT architecture employs two types of attention: self-attention, where 3D latent tokens interact with each other to establish spatial structure, and cross-attention, where 3D tokens query 2D image patch features for visual conditioning. As shown in the upper part of Fig. 2, we inject temporal information through both pathways with complementary roles: cross-attention propagates *visual memory*, retaining image-conditioned appearance from earlier observations, while self-attention propagates *structural memory*, maintaining consistency of the 3D latent representation over time. Both paths maintain importance-scored KV caches with bounded capacity, updated as frames arrive sequentially.

Cross-Attention Visual Memory In the standard pipeline, cross-attention conditions the 3D latent exclusively on the current frame's image patches. As

the object's visible appearance changes across frames—due to object motion or camera movement—previously observed regions may no longer appear in the current view, and the model is forced to hallucinate their appearance solely from the current observation. By caching image patch features from earlier frames, we enable the model to recall visual information from different observations, preserving appearance consistency for regions that were once visible but are currently occluded.

We maintain a KV pool $\mathcal{P}^{\text{cross}}$ with bounded capacity N_{cross} for 2D image patch tokens from historical frames. During cross-attention, the cached patches are concatenated with the current frame's:

$$K = [K_{\text{img}};\ K_{\text{pool}}], \quad V = [V_{\text{img}};\ V_{\text{pool}}]. \tag{1}$$

The 3D tokens can now attend to both current and cached patches, naturally fusing multi-temporal visual information when the current view lacks certain regions.

The pool is managed through two complementary criteria. For eviction, each cached patch is scored by *complementarity*—its dissimilarity to current-frame patches, ensuring we retain information absent from the current observation—combined with *usage*—the attention it receives from 3D tokens during the current generation, ensuring the retained information remains relevant. This combination addresses a subtle failure mode: complementarity alone would indefinitely retain outdated patches whose content has since changed but still appears "unique"; usage provides a necessary check that the cached visual information is still actively leveraged. For admission, current-frame patches most dissimilar to the remaining pool are selected, maintaining visual diversity across the cache.

We validate this design in Fig. 3: a construction worker faces the camera at T_0 then turns away at T_1–T_3, with distinct front/back jacket textures. Without cross-attention memory (b), the model generates later frames using only back-view patches, producing incorrect front-face textures when rendered from the *front view*. With cross-attention memory (c), cached patches from T_0 retain the observed front appearance and produce correct textures.

Beyond its role in visual memory, the cross-attention mechanism provides a natural measure of observation confidence for each 3D token. We compute the normalized entropy of each token's cross-attention distribution:

$$H(l) = -\frac{1}{\log P} \sum_{p} a_l^{(p)} \log a_l^{(p)}, \tag{2}$$

where $a_l^{(p)}$ is the attention weight from 3D token l to image patch p, and P is the total number of patches (current frame plus cached). When a token corresponds to a region directly visible in the input image, it attends to a small number of relevant patches with high specificity, producing low entropy. When it lies in an occluded region with no direct image correspondence, the model must aggregate diffuse context across the entire image, yielding high entropy (Fig. 4(a)). This cross-attention entropy thus serves as an implicit indicator of whether a 3D

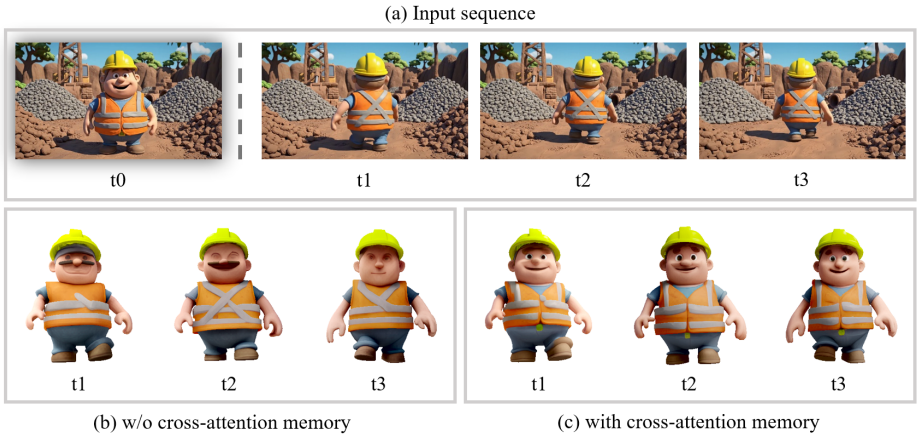

Fig. 3: Effect of Cross-Attention Visual Memory. **(a)** T_0: front view; T_1–T_3: worker turns away. The jacket has asymmetric front/back textures. **(b)** Without cross-attention memory, *front-view* renderings show hallucinated, incorrect textures. **(c)** With cross-attention memory, cached patches from T_0 preserve the observed front texture.

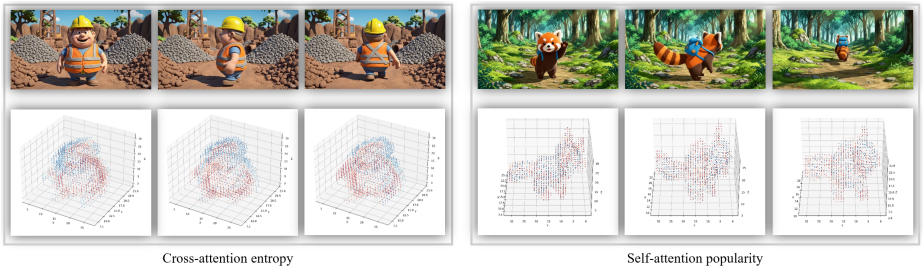

Fig. 4: Visualization of attention-based importance metrics. **(a) Cross-Attention Entropy:** blue = low entropy (visible, confident); red = high entropy (occluded, hallucinated). **(b) Self-Attention Popularity:** red = structurally important (edges, joints); blue = flat, uniform regions.

token is reliably grounded in observation or speculatively hallucinated—a signal we leverage for self-attention cache management in the following section.

Self-Attention Temporal Memory In independent per-frame generation, each frame's self-attention mechanism builds the 3D latent structure from scratch. Since the generation model can only observe the visible portion of the object in each frame, it must hallucinate content for unobserved regions—and these hallucinations vary arbitrarily across frames, causing structural discontinuities. By propagating self-attention KV pairs from previous frames, we provide a structural prior that anchors the current frame's 3D representation to earlier generations, encouraging consistent geometry even in unobserved regions.

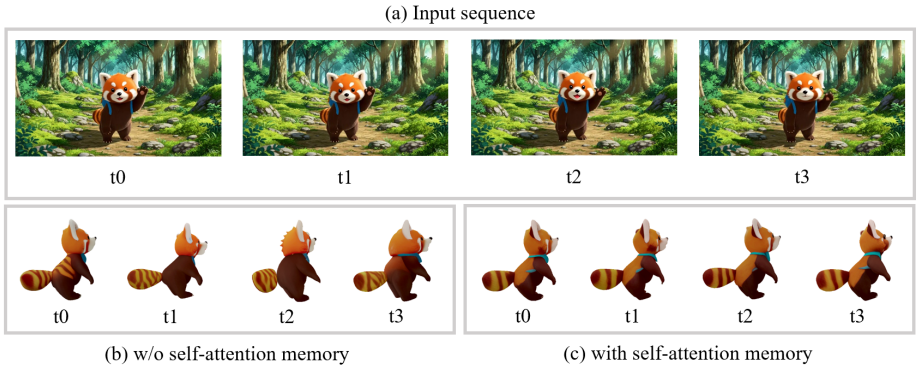

Fig. 5: Effect of Self-Attention Temporal Memory. **(a)** Four consecutive frontal-view input frames of a red panda waving. **(b)** Without self-attention memory, *side-view* renderings reveal severe inconsistencies in unobserved regions. **(c)** With self-attention memory, cached structural KV pairs produce consistent side-view appearance across frames.

We maintain a KV pool $\mathcal{P}^{\text{self}}$ with bounded capacity N_{self} for 3D latent tokens from historical frames. During self-attention, the cached KV pairs are concatenated with the current frame's tokens:

$$K = [K_{\text{curr}};\ K_{\text{pool}}], \quad V = [V_{\text{curr}};\ V_{\text{pool}}]. \tag{3}$$

Queries come only from the current frame's tokens, so they can attend to historical structural information—analogous to temporal self-attention in video transformers [8], but achieved training-free through KV concatenation.

The pool is managed by scoring each token—both existing and newly generated—through three complementary signals: (1) *Confidence* $C(l) = 1 - H(l)$, derived from the cross-attention entropy introduced above, prioritizing tokens from visible, reliably generated regions as structural anchors; (2) *Popularity*, the attention a token receives from other tokens during generation, identifying structurally salient "hub" tokens at geometric boundaries and articulation points (Fig. 4(b)); and (3) *Temporal decay* $\exp(-\lambda_{\text{decay}} \cdot \Delta t)$, gradually discounting older tokens to allow the pool to adapt. All tokens compete under a unified importance score:

$$S^{\text{self}}(l) = C(l) \cdot \text{Pop}(l) \cdot \exp(-\lambda_{\text{decay}} \cdot \Delta t), \tag{4}$$

and the pool retains the top-scoring tokens up to its capacity.

We validate this design in Fig. 5: four nearly identical consecutive frontal-view frames of a red panda are rendered from a *side view* to expose the unobserved regions. Without self-attention memory (b), each frame's hallucination differs drastically; with it (c), the structural prior from cached KV pairs produces consistent side-view geometry and appearance.

3.3 Gradient-Guided Velocity Correction

Attention-level memory biases the generation toward temporal coherence through shared representational context, but it provides only implicit, soft coupling—the model is encouraged, not constrained, to maintain consistency. Moreover, this internal mechanism offers no channel through which external signals, such as observation-derived geometric or perceptual constraints, can be directly incorporated. A complementary form of guidance is therefore needed: one that can explicitly shape the generation trajectory by injecting differentiable objectives into the sampling process.

To this end, we apply gradient-based guidance at the ODE solver level (Fig. 2, bottom). At each step s, we obtain a lookahead estimate $\hat{x}_0 = x_s - s \cdot v_s$ and correct the velocity via:

$$v_{\text{guided}} = v_s + \alpha_s \cdot \nabla_{\hat{x}_0} \mathcal{L}_{\text{guide}}, \tag{5}$$

where α_s is a step-dependent guidance strength. The guidance loss decomposes into two components: Temporal Guidance for trajectory smoothness and Alignment Guidance for observation consistency.

Temporal Guidance. We encourage the predicted clean output to remain close to the previous frame's prediction in latent space:

$$\mathcal{L}_{\text{traj}} = \|\hat{x}_0^{(t)} - \hat{x}_0^{(t-1)}\|^2, \tag{6}$$

where $\hat{x}_0^{(t-1)}$ is stored from the previous frame's corresponding ODE step. This lightweight loss operates entirely in latent space and is applied at every step, acting as a soft anchor that prevents the sampling trajectory from drifting across frames.

Alignment Guidance. Alignment Guidance ensures consistency with observations through two complementary signals, each applied at the stage where it is most effective.

In Stage 1, where coarse geometry and layout pose are jointly determined, we apply geometric alignment to steer the object's spatial arrangement toward the observed point map. We decode the lookahead estimate through the sparse structure decoder to obtain approximate voxel positions $\tilde{\mathcal{V}}$ and the intermediate pose $T_t = (s_t, R_t, \mathbf{t}_t)$, then compute a one-directional Chamfer distance from the point map $\mathcal{P}_t$ to the placed structure:

$$\mathcal{L}_{\text{geo}} = \frac{1}{|\mathcal{P}_t|} \sum_{p \in \mathcal{P}_t} \min_{q \in T_t(\tilde{\mathcal{V}})} \|p - q\|^2. \tag{7}$$

We use a one-directional distance—from point map to generated structure—because the single-view point map covers only the visible portion of the object; a bidirectional distance would incorrectly penalize valid but unobserved geometry.

In Stage 2, where detailed geometry and texture are refined upon the structure from Stage 1, we apply rendering alignment. We decode the lookahead

estimate and render it via a differentiable renderer, computing a perceptual loss:

$$\begin{aligned}\mathcal{L}_{\text{render}} = \mathcal{L}_{\text{LPIPS}}\Big(\text{render}(\hat{x}_0^{(t)}),\, I_t\Big) \\ + \lambda_{\text{temp}} \cdot \mathcal{L}_{\text{LPIPS}}\Big(\text{render}(\hat{x}_0^{(t)}),\, \text{render}(\hat{x}_0^{(t-1)})\Big)\,.\end{aligned} \tag{8}$$

The first term ensures fidelity to the current input image, and the second encourages perceptual consistency between consecutive frames' renderings.

The total guidance combines all three components:

$$\mathcal{L}_{\text{guide}} = \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{render}}\mathcal{L}_{\text{render}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}}. \tag{9}$$

In practice, $\mathcal{L}_{\text{traj}}$ is applied at every ODE step in both stages. $\mathcal{L}_{\text{geo}}$ is applied only in Stage 1 at later steps, where the latent is sufficiently denoised for meaningful decoding. $\mathcal{L}_{\text{render}}$ is applied only in Stage 2 at selected steps.

3.4 HexPlane Temporal Smoothing

After online generation of all frames, we apply a post-processing step that exploits the global temporal structure (Fig. 2, right). The online mechanisms operate causally and may leave residual high-frequency temporal noise.

For latent feature smoothing, we factorize the sequence of per-frame latent features into six learnable 2D planes via HexPlane decomposition [2]: three spatial planes (XY, XZ, YZ) capturing shared 3D structure, and three spatiotemporal planes (XT, YT, ZT) modeling temporal variations. This low-rank decomposition acts as an implicit temporal filter—frame-to-frame fluctuations that do not form coherent spatiotemporal patterns are naturally suppressed. The optimization prioritizes faithful reconstruction of per-frame features while a mild temporal regularization further encourages coherence across adjacent frames. The smoothed features replace the originals before final decoding.

For pose refinement, we jointly optimize the pose sequence $\{T_t\}$ to balance spatial alignment with temporal smoothness. An alignment term minimizes the unidirectional Chamfer distance from the observed point map to the transformed generated object at each frame, ensuring accurate scene placement. A smoothness term penalizes abrupt changes in scale, rotation, and translation between consecutive frames. This global optimization over all frames complements the causal online guidance (Sec. 3.3) by correcting residual drift with access to the full temporal context.

4 Experiments

4.1 Experimental Setup

All experiments are conducted on a single NVIDIA A100 (80GB) GPU. We follow SAM3D's default configuration for the base generation pipeline (50 ODE steps for Stage 1, 25 for Stage 2). For Tempo-SAM3D, the self-attention KV

Table 1: Quantitative comparison following V2M4 evaluation. Best in **bold**. (*Simple*: Consistent4D; *Complex*: Mixamo & Sketchfab.)

	Method	CLIP↑	LPIPS↓	FVD↓	DreamSim↓
Simple	TRELLIS	0.8905	0.1597	1342.66	0.1282
	DreamMesh4D	0.8692	0.1019	914.28	0.0937
	V2M4	0.9259	0.1017	**825.59**	**0.0688**
	Tempo-SAM3D (Ours)	**0.9312**	**0.0992**	841.27	0.0702
Complex	TRELLIS	0.8887	0.1265	1216.19	0.1492
	DreamMesh4D	0.8256	0.0804	1079.02	0.1850
	V2M4	0.9008	0.0747	666.04	0.1220
	Tempo-SAM3D (Ours)	**0.9085**	**0.0728**	**642.38**	**0.1165**

cache capacity is $N_{\text{self}} = 8000$ (Stage 1) and 15000 (Stage 2); the cross-attention cache capacity is $N_{\text{cross}} = 1024$. In the importance scoring, the temporal decay rate is $\lambda_{\text{decay}} = 0.1$. The guidance strengths are $\lambda_{\text{traj}} = 0.3$ with linear decay schedule, $\lambda_{\text{geo}} = 0.05$, and $\lambda_{\text{render}} = 0.1$. Geometric guidance is applied at the last 50% of Stage 1 steps; rendering guidance is applied at the last 50% of Stage 2 steps. For HexPlane smoothing, the temporal resolution is set to 48 with plane dimension 32.

Following V2M4 [3], we evaluate on 40 animation videos from two sources: 20 sequences from Consistent4D [12] featuring simple objects with subtle movements (denoted *Simple*), and 20 sequences from Mixamo [25] and Sketchfab [30] featuring complex objects with large-scale movements (denoted *Complex*). These videos have white backgrounds with pre-segmented foreground objects. In addition, we construct the Tempo-SAM3D dataset comprising 20 video sequences generated by a video generation model, featuring diverse dynamic subjects with natural backgrounds. For each sequence, we recover camera trajectories and per-frame dynamic point maps through 3D/4D reconstruction, providing accurate spatial references for both generation conditioning and evaluation. This dataset requires spatially-grounded generation—the generated objects must be correctly positioned within the observed scene—enabling evaluation of both generation quality and spatial accuracy.

On the above benchmarks, we compare against TRELLIS, DreamMesh4D, V2M4, and SAM3D [3, 4, 21, 40]. We evaluate rendering quality, temporal consistency, and spatial positioning accuracy.

4.2 Monocular Video to 4D Generation

Tab. 1 presents quantitative comparisons on the V2M4 benchmark, where we follow V2M4's evaluation protocol and report baseline results from the respective publications alongside ours. Tempo-SAM3D achieves competitive or superior results across both subsets, with particularly strong performance on Complex sequences involving large-scale motions. Notably, our method produces these results without per-scene training or multi-stage optimization, in contrast to DreamMesh4D and V2M4.

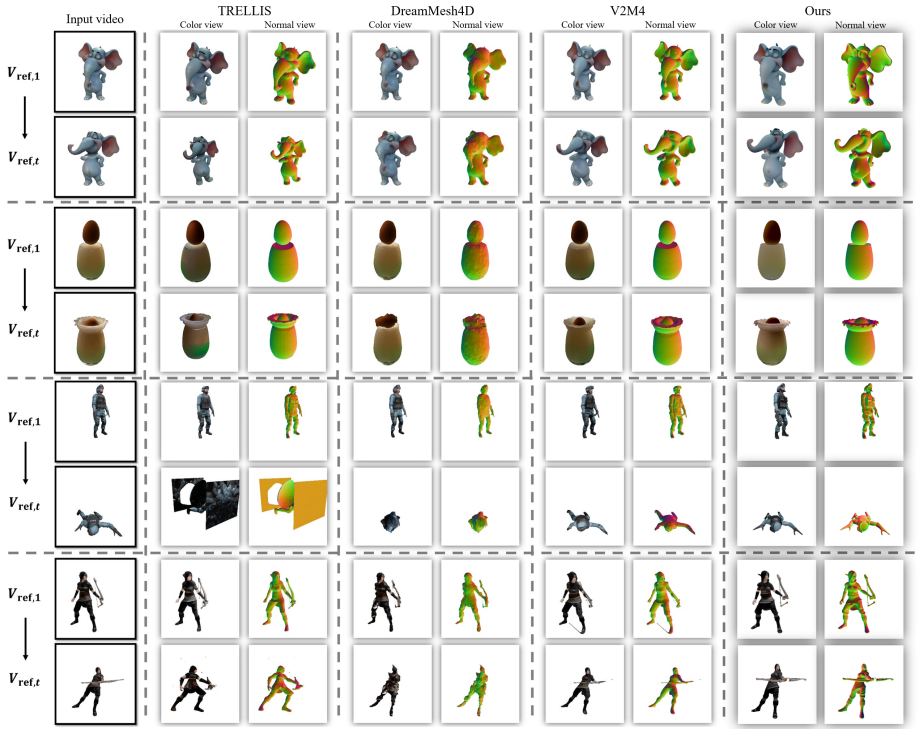

Fig. 6: Qualitative comparison following V2M4 evaluation. For each sequence, two frames are rendered as RGB and Normal. Our method produces sharper geometry and more faithful textures.

Tab. 2 evaluates on the Tempo-SAM3D dataset, which features videos with natural backgrounds requiring spatially-grounded generation. TRELLIS and V2M4, lacking layout awareness, are excluded from spatial positioning metrics. Tempo-SAM3D achieves the best results across all evaluated aspects, including rendering quality, temporal consistency, and spatial positioning accuracy.

Fig. 6 presents qualitative comparisons on representative V2M4 benchmark sequences, following the same evaluation setting. For each of four sequences, we show two frames rendered as RGB and Normal maps. Tempo-SAM3D produces sharper geometric details and more faithful textures that better match the input appearance.

Table 2: Quantitative comparison on the Tempo-SAM3D dataset. "—": not applicable.

Method	LPIPS↓	PSNR↑	FVD↓	IoU↑	D-RMSE↓	ACC_{5cm} ↑
TRELLIS	0.165	17.82	1456.28	—	—	—
V2M4	0.138	18.76	781.56	—	—	—
SAM3D	0.128	19.45	1218.63	73.5	0.028	90.3
Tempo-SAM3D (Ours)	**0.119**	**20.12**	**732.45**	**83.8**	**0.018**	**95.8**

Fig. 7: Qualitative gallery on four sequences from the Tempo-SAM3D dataset. For each sequence: Row 1 shows input frames; Row 2 shows generated objects placed within the scene point map; Row 3 shows per-frame RGB appearance and surface Normal geometry.

Table 3: Progressive ablation on the Tempo-SAM3D dataset.

Configuration	FVD↓	LPIPS↓	PSNR↑	IoU↑	D-RMSE↓	ACC_{5cm} ↑
(a) Baseline (SAM3D)	1218.63	0.128	19.45	73.5	0.028	90.3
(b) + Self-Attn Memory	1076.52	0.131	19.32	73.1	0.029	89.8
(c) + Temporal Guidance	935.18	0.130	19.38	73.3	0.028	90.1
(d) + Alignment Guidance	871.25	0.121	**20.18**	82.6	0.019	95.2
(e) + HexPlane (**Full**)	**732.45**	**0.119**	20.12	**83.8**	**0.018**	**95.8**

Fig. 7 presents qualitative results on four diverse sequences from the Tempo-SAM3D dataset. For each sequence, we show the input frames, the generated objects placed within the scene, and close-up views of per-frame appearance and geometry. The in-scene placement (Row 2) highlights a distinctive advantage of Tempo-SAM3D: unlike canonical-space 4D generation methods that produce objects in isolation, our approach generates spatially-grounded 4D content where each object is precisely positioned within the observed scene.

4.3 Ablation Studies

We conduct ablation studies on the Tempo-SAM3D dataset to evaluate the contribution of each component. Tab. 3 reports a progressive ablation where we add one component at a time, measuring temporal consistency, rendering quality, and spatial positioning accuracy.

Self-attention memory (b) improves temporal consistency by propagating 3D structural memory across frames, as visualized in Fig. 5. Temporal guidance (c) further smooths the generation trajectory, yielding continued FVD reduction. Alignment guidance (d) brings the largest improvement in spatial positioning and rendering quality by directly aligning the generation with observations through geometric and rendering constraints. HexPlane smoothing (e) provides a global temporal polish, further reducing FVD and refining spatial accuracy through joint pose optimization.

Note that cross-attention visual memory is not included in the progressive ablation above, as its primary benefit—preserving appearance from previously observed but currently occluded regions—is not well captured by metrics evaluated at the input view. We instead provide a dedicated qualitative ablation in Fig. 3. In the ablation, a construction worker's front-face texture is observed at T_0 but becomes invisible as the worker turns away at T_1–T_3. Without cross-attention memory, the model forgets the front appearance and hallucinates incorrect textures; with it, cached image patches from T_0 preserve the front-face appearance, demonstrating the value of visual memory for maintaining appearance consistency as the object's visible appearance changes.

5 Conclusion

We have presented Tempo-SAM3D, a training-free framework that extends layout-aware 3D generation to temporally coherent 4D generation from monoc-

ular video. Our approach addresses temporal inconsistency at three complementary levels: Dual-Path Temporal Attention propagates visual and structural memory via importance-driven KV caches in cross-attention and self-attention; gradient-guided velocity correction enforces trajectory smoothness and observation alignment at the ODE solver level; and HexPlane-based temporal smoothing applies global spatiotemporal regularization as a post-processing step. Together, these mechanisms form a comprehensive consistency pipeline without modifying the underlying model's weights. Experiments on multiple datasets demonstrate that Tempo-SAM3D achieves clear improvements across rendering quality, temporal consistency, and spatial accuracy, while uniquely producing spatially-grounded 4D content that preserves layout awareness. We believe the principle of injecting multi-level temporal consistency into per-frame generative pipelines is broadly applicable to other architectures and modalities beyond 3D generation.

Acknowledgement. We thank the anonymous reviewers and area chairs for their constructive feedback and helpful suggestions. This work is supported by the NFS, China (U22A2061), and 230601GP0004.

References

1. Boss, M., Huang, Z., Vasishta, A., Jampani, V.: SF3D: Stable fast 3D mesh reconstruction with UV-unwrapping and illumination disentanglement. In: CVPR (2025)
2. Cao, A., Johnson, J.: HexPlane: A fast representation for dynamic scenes. In: CVPR (2023)
3. Chen, J., Zhang, B., Tang, X., Wonka, P.: V2M4: 4D mesh animation reconstruction from a single monocular video. In: ICCV (2025)
4. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: SAM 3D: 3Dfy anything in images. arXiv preprint arXiv:2511.16624 (2025)
5. Cong, Z., Zhao, Q., Jeon, M., Tulsiani, S.: Flow3R: Factored flow prediction for scalable visual geometry learning. arXiv preprint arXiv:2602.20157 (2026)
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: ECCV (2014)
7. Fridovich-Keil, S., Meanti, G., Warburg, F., Recht, B., Kanazawa, A.: K-Planes: Explicit radiance fields in space, time, and appearance. In: CVPR (2023)
8. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In: ICLR (2024)
9. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: LRM: Large reconstruction model for single image to 3D. In: ICLR (2024)
10. Huang, Y.H., Sun, Y.T., Yang, Z., Lyu, X., Cao, Y.P., Qi, X.: SC-GS: Sparse-controlled Gaussian splatting for editable dynamic scenes. In: CVPR (2024)
11. Huang, Z., Boss, M., Vasishta, A., Rehg, J.M., Jampani, V.: SPAR3D: Stable point-aware reconstruction of 3D objects from single images. In: CVPR (2025)

12. Jiang, Y., Zhang, L., Gao, J., Hu, W., Yao, Y.: Consistent4D: Consistent 360° dynamic object generation from monocular video. arXiv preprint arXiv:2311.02848 (2023)
13. Karhade, J., Keetha, N., Zhang, Y., Gupta, T., Sharma, A., Scherer, S., Ramanan, D.: Any4D: Unified feed-forward metric 4D reconstruction. arXiv preprint arXiv:2512.10935 (2025)
14. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM. In: CVPR (2024)
15. Kratimenos, A., Lei, J., Daniilidis, K.: DynMF: Neural motion factorization for real-time dynamic view synthesis with 3D Gaussian splatting. In: ECCV (2024)
16. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3D with MASt3R. In: ECCV (2024)
17. Li, B., Yan, Z., Wu, D., Jiang, H., Zha, H.: Learn to memorize and to forget: A continual learning perspective of dynamic SLAM. In: ECCV (2024)
18. Li, B., Yan, Z., Wu, D., Zha, H.: Proactive scene decomposition and reconstruction. In: ICCV (2025)
19. Li, W., Liu, J., Yan, H., Chen, R., Liang, Y., Chen, X., Tan, P., Long, X.: CraftsMan3D: High-fidelity mesh generation with 3D native diffusion and interactive geometry refiner. In: CVPR (2025)
20. Li, Y., Zou, Z.X., Liu, Z., Wang, D., Liang, Y., Yu, Z., Liu, X., Guo, Y.C., Liang, D., Ouyang, W., et al.: TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. IEEE Trans. Pattern Anal. Mach. Intell. (2025)
21. Li, Z., Chen, Y., Liu, P.: DreamMesh4D: Video-to-4D generation with sparse-controlled Gaussian-mesh hybrid representation. In: NeurIPS (2024)
22. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth Anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
23. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
24. Matsuki, H., Murai, R., Kelly, P.H.J., Davison, A.J.: Gaussian splatting SLAM. In: CVPR (2024)
25. Mixamo: Mixamo (2025), `https://www.mixamo.com`, accessed: 30 June 2026
26. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: A versatile and accurate monocular SLAM system. IEEE Trans. Robotics **31**(5), 1147–1163 (2015)
27. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: Neural radiance fields for dynamic scenes. In: CVPR (2021)
28. Ren, J., Pan, L., Tang, J., Zhang, C., Cao, A., Zeng, G., Liu, Z.: DreamGaussian4D: Generative 4D Gaussian splatting. arXiv preprint arXiv:2312.17142 (2023)
29. Ren, J., Xie, C., Mirzaei, A., Kreis, K., Liu, Z., Torralba, A., Fidler, S., Kim, S.W., Ling, H., et al.: L4GM: Large 4D Gaussian reconstruction model. In: NeurIPS (2024)
30. Sketchfab: Sketchfab (2025), `https://www.sketchfab.com`, accessed: 30 June 2026
31. Teed, Z., Deng, J.: DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In: NeurIPS (2021)
32. Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., Cao, Y.P.: TripoSR: Fast 3D object reconstruction from a single image. arXiv preprint arXiv:2403.02151 (2024)
33. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: CVPR (2025)

34. Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., Kanazawa, A.: Shape of motion: 4D reconstruction from a single video. In: ICCV (2025)
35. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: CVPR (2025)
36. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3D vision made easy. In: CVPR (2024)
37. Wang, Z., Wang, Y., Chen, Y., Xiang, C., Chen, S., Yu, D., Li, C., Su, H., Zhu, J.: CRM: Single image to 3D textured mesh with convolutional reconstruction model. In: ECCV (2024)
38. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4D Gaussian Splatting for real-time dynamic scene rendering. In: CVPR (2024)
39. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3D: Scalable image-to-3D generation via 3D latent diffusion transformer. In: NeurIPS (2024)
40. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: TRELLIS: Structured 3D latents for scalable and versatile 3D generation. In: CVPR (2025)
41. Xie, Y., Yao, C.H., Voleti, V., Jiang, H., Jampani, V.: SV4D: Dynamic 3D content generation with multi-frame and multi-view consistency. arXiv preprint arXiv:2407.17470 (2024)
42. Xu, J., Cheng, W., Gao, Y., Wang, X., Gao, S., Shan, Y.: InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. arXiv preprint arXiv:2404.07191 (2024)
43. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In: CVPR (2025)
44. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything: Unleashing the power of large-scale unlabeled data. In: CVPR (2024)
45. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything V2. In: NeurIPS (2024)
46. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In: CVPR (2024)
47. Yin, Y., Xu, D., Wang, Z., Zhao, Y., Wei, Y.: 4DGen: Grounded 4D content generation with spatial-temporal consistency. arXiv preprint arXiv:2312.17225 (2023)
48. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-SLAM: Photo-realistic dense SLAM with Gaussian splatting. In: CVPR (2024)
49. Zeng, Y., Jiang, Y., Zhu, S., Lu, Y., Lin, Y., Zhu, H., Hu, W., Cao, X., Yao, Y.: STAG4D: Spatial-temporal anchored generative 4D Gaussians. In: ECCV (2024)
50. Zhang, H., Chen, X., Wang, Y., Liu, X., Wang, Y., Qiao, Y.: 4Diffusion: Multi-view video diffusion model for 4D generation. In: NeurIPS (2024)
51. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: MonST3R: A simple approach for estimating geometry in the presence of motion. arXiv preprint arXiv:2410.03825 (2024)
52. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3D assets generation. arXiv preprint arXiv:2501.12202 (2025)
53. Zhu, H., He, T., Yu, X., Guo, J., Chen, Z., Bian, J.: AR4D: Autoregressive 4D generation from monocular videos. arXiv preprint arXiv:2501.01722 (2025)