ANYSPACETIME: FEED-FORWARD SPACETIME NOVEL VIEW SYNTHESIS FROM ARBITRARY OBSERVATIONS

Anonymous authors
Paper under double-blind review



Figure 1: **AnySpaceTime provides a unified framework for spacetime novel view synthesis from arbitrary observations.** Given reference observations arbitrarily distributed across time and viewpoint, together with their timestamps and camera parameters, AnySpaceTime synthesizes a queried target image in a single feed-forward pass. **The target time and viewpoint are specified independently**, as illustrated by the query at (T_i, V_j).

ABSTRACT

Novel view synthesis (NVS) has progressed significantly, but existing settings are typically tailored to specific observation patterns, such as static multi-view scenes, dynamic videos, or synchronized multi-camera captures. This paper introduces **Space-Time NVS**, a unified task that takes timestamped and posed images distributed across time and viewpoint as input and synthesizes images at a target viewpoint and a target time within the observed temporal interval, subsuming the above settings as special cases. We propose **AnySpaceTime**, an end-to-end feed-forward framework for this task that supports independently specified time and view queries in a single forward pass. Our key idea is to decouple spatial representation from temporal adaptation: we freeze a pretrained NVS encoder to preserve strong spatial priors and introduce a lightweight motion branch that predicts time-dependent updates in the latent space. The target time determines the latent update, while the target camera pose determines rendering, with static scenes handled as the shared-state special case. This design handles static multi-view inputs, dynamic videos, and asynchronous multi-camera observations without requiring a fixed camera layout, synchronized or contiguous frames, or supervision from depth, optical flow, or 3D motion. It generalizes to unseen scenes and irregular observations, and experiments across multiple datasets.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Novel View Synthesis (NVS) renders unseen views of a scene from a set of reference images. Feed-forward NVS models generalize to unseen scenes without per-scene optimization, by predicting either explicit 3D representations (Charatan et al., 2024; Chen et al., 2024) or latent scene representations that are decoded directly into images (Jin et al., 2025; Szymanowicz et al., 2026). However, they assume that all reference images share the same static scene state. Dynamic scenes, where appearance changes with both camera and scene motion (Li et al., 2021; Pumarola et al., 2020), are instead handled by methods designed for specific capture setups, such as monocular videos (Liang et al., 2024; Lin et al., 2026a) or synchronized multi-camera rigs in driving scenarios (Yang et al., 2025; Shi et al., 2026). Since these setups differ in camera layout, temporal sampling, and synchronization, models developed for one do not readily transfer to another. In this work, we ask whether a single feed-forward model can handle both static and dynamic scenes under diverse observation patterns, while synthesizing views at independently specified target viewpoints and times.

To this end, we introduce **Space-Time Novel View Synthesis** (Fig. 1), which renders a target view at a queried camera and time from a set of posed, timestamped reference images. The main obstacle to unification is that each capture setting implies a different input structure: a static multi-view capture observes many viewpoints at one time, a monocular video couples viewpoint and time along a single trajectory, and multi-camera rigs observe several viewpoints at synchronized or unsynchronized times. We instead describe each reference solely by its image, timestamp, and camera parameters, forming a **Unified Reference Set** in which each setting reduces to a sampling pattern over viewpoint and time, with static scenes as the special case of identical timestamps. Yet this unification exposes the central difficulty of the task: *observations that complement each other spatially may conflict temporally*. For example, a side view may reveal parts of a moving person that are occluded in the frontal view, yet capture the person in a different pose.

To understand how such conflicts can be resolved, we apply a pretrained static NVS model to mixed-time references (Fig. 2). At novel viewpoints, static backgrounds are rendered sharply while moving subjects exhibit ghosting: the spatial prior is still effective, but the model has no temporal signal with which to select a coherent dynamic state. In contrast, when the target camera coincides with or lies near a source camera, the model recovers the state observed at that source time, i.e., **source-time dynamic states remain locally renderable**. This leads to our key insight: *a pretrained static NVS representation already encodes each source-time state in a renderable form, anchored to the features of its source reference, so dynamic NVS can be cast as adapting these features to the target time before rendering, rather than learning a spatiotemporal representation from scratch.*



Figure 2: **The static-scene feed-forward model applied to dynamic observations.** Observations captured in a dynamic scene are fed into LagerNVS (Szymanowicz et al., 2026), a feed-forward model trained under a static-scene assumption. Static regions remain sharp, while moving people exhibit severe ghosting caused by temporal inconsistency.

Based on this insight, we propose **AnySpace-Time**, a feed-forward framework for latent temporal adaptation. We keep the spatial encoder of the pretrained static NVS model frozen to preserve its spatial prior, and add a lightweight motion branch that adapts the encoded features to the target time. The motion branch extracts a compact set of scene-dependent latent motion basis from the references and, for each reference token, predicts coefficients conditioned on the source and target times that combine these basis into a residual feature update. Following prior motion-factorization models (Kratimenos et al., 2024; Wang et al., 2025), the shared basis constrain the updates to a low-dimensional space, while the spatially varying coefficients allow different regions to follow different temporal changes. A viewpoint-conditioned decoder, adapted with LoRA, then renders the updated features from the target camera. Because time and viewpoint enter the model separately, the updated features can be computed once per target time and reused across viewpoints. Combined with the Unified Reference Set, a single model thus

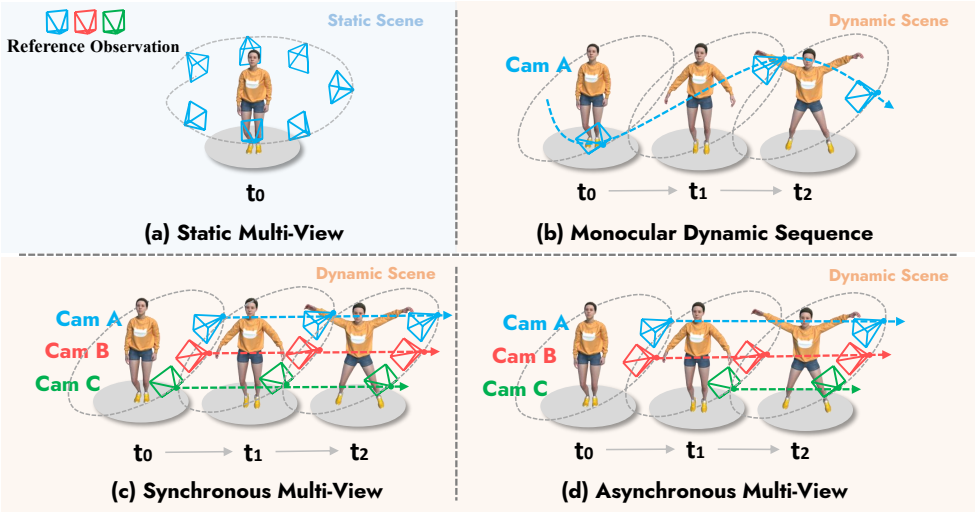

Figure 3: **Representative space-time sampling patterns of reference observations.** The four panels illustrate static multi-view observations, monocular dynamic sequences, synchronized multi-view observations, and asynchronous multi-view observations. Each reference image is associated with a timestamp and camera parameters, allowing all four settings to share the same reference-set representation. The illustrated patterns are representative rather than exhaustive.

handles static and dynamic scenes across the sampling patterns in Fig. 3 without setting-specific fine-tuning; it is trained with RGB supervision only and requires no test-time optimization.

In summary, this work makes the following contributions:

- We formulate **Space-Time NVS** with a unified reference set that represents diverse capture settings as posed and timestamped observations sampled over viewpoint and time, enabling independent target-viewpoint and target-time queries.
- We propose **AnySpaceTime**, a feed-forward framework that introduces temporal adaptation directly into the renderable latent space of a pretrained NVS model, preserving its learned spatial prior while extending it to both static and dynamic scenes.
- We introduce a **factorized temporal representation** for modeling dynamic object motion, combining scene-dependent shared motion bases with source-local, token-wise coefficients conditioned on the target time. This design enables a single model to handle diverse static and dynamic observation patterns without setting-specific fine-tuning.

2 Related Work

Feed-forward novel view synthesis. Feed-forward NVS synthesizes novel views of unseen scenes without per-scene optimization, using either explicit 3D representations (Charatan et al., 2024; Chen et al., 2024) or learned renderable features (Jin et al., 2025; Jiang et al., 2025; Szymanowicz et al., 2026). These methods primarily assume a static scene state. Recent approaches extend feed-forward NVS to dynamic scenes through time-varying Gaussian representations (Shen et al., 2025; Lin et al., 2026a; 2025), sometimes with additional image-space synthesis or refinement (Liang et al., 2024; Kim et al., 2026). Such methods commonly rely on explicit dynamic 3D representations, geometric or motion supervision, or auxiliary generation stages. AnySpaceTime instead preserves a pretrained spatial representation and learns target-time adaptation directly in its renderable latent space using only image supervision.

Dynamic and feed-forward 4D representations. Dynamic scene representations often model temporal variation through canonical deformation or shared motion structures (Pumarola et al., 2020; Wu et al., 2024; Kratimenos et al., 2024; Wang et al., 2025), but typically require per-scene optimization. Recent feed-forward 4D methods instead predict geometry and motion directly, including queried 3D points or relative motion (Zhang et al., 2026; Luo et al., 2026), dense trajectories (Jiang et al., 2026), and structured motion bases (Lin et al., 2026b). Their primary outputs are geometric quantities rather than directly renderable appearance representations. AnySpaceTime similarly exploits shared motion structure, but introduces it directly into a pretrained NVS latent space, enabling target-time feature adaptation and direct novel-view rendering without explicit trajectory prediction.

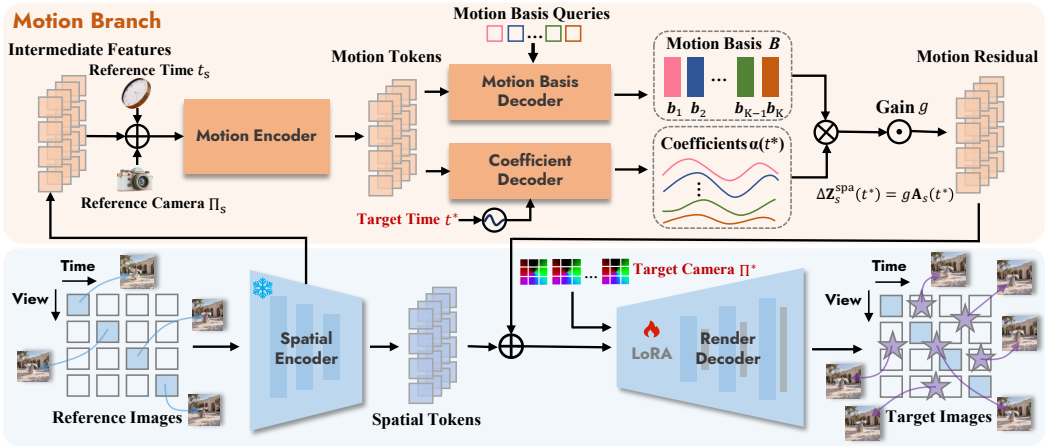

Figure 4: **AnySpaceTime model architecture.** A frozen spatial encoder extracts spatial tokens and intermediate features from the reference images and cameras. The motion encoder combines the intermediate features with reference-ray and time embeddings to produce motion tokens. Learnable queries extract shared motion bases $\mathbf{B}$, while a target-time-conditioned coefficient decoder predicts reference-anchored coefficients $\mathbf{A}_s(t^*)$. Their product forms a motion residual $g\mathbf{A}_s(t^*)\mathbf{B}$, which updates the spatial tokens. Conditioned on the target camera Π^*, the LoRA-adapted render decoder synthesizes the target image.

Overall, existing methods provide transferable spatial priors, explicit 4D motion modeling, or flexible spatiotemporal generation, but these capabilities are largely developed separately. To bridge this gap, we propose AnySpaceTime, which adapts a pretrained NVS latent representation to the target time and directly renders the target view from posed and timestamped observations, without explicit dynamic geometry, trajectory prediction, or iterative generation.

3 Method

We propose **AnySpaceTime**, a feed-forward framework for synthesizing a target image from posed, timestamped references at a specified time and viewpoint. As shown in Fig. 4, a frozen pretrained NVS encoder extracts spatial tokens and intermediate features from the reference set. A lightweight motion branch aggregates the intermediate features and factorizes temporal variation into scene-shared motion bases and reference-anchored coefficients conditioned on the target time. Their weighted combination produces a motion residual that adapts the spatial tokens to the target-time scene state. A LoRA-adapted render decoder (Hu et al., 2022), conditioned on the target camera then synthesizes the image from the adapted tokens.

3.1 Space-Time NVS Formulation

We represent static multi-view captures, monocular dynamic sequences, and synchronous or asynchronous multi-view observations using our **Unified Reference Set** $\mathcal{R}$ (Fig. 3):

$$\mathcal{R} = \{(\mathbf{I}_s, t_s, \Pi_s)\}_{s=1}^{S}, \tag{1}$$

where $\mathbf{I}_s$, t_s, and Π_s denote the RGB image, timestamp, and camera parameters of observation s respectively. Given $\mathcal{R}$ and a target query specifying time t^* and camera parameters Π^*, our model $\mathcal{M}_\theta$ with trainable parameters θ predicts

$$\hat{I}^* = \mathcal{M}_\theta(\mathcal{R}, t^*, \Pi^*), \tag{2}$$

where I^* denotes the corresponding ground-truth target image. For dynamic scenes with a nonzero reference time span, we map the reference timestamps to $[0, 1]$ using their $\min/\max$ values and specify $t^* \in [0, 1]$ on the same scale; otherwise, we set $t_s = t^* = 0$.

3.2 Spatial and Motion Encoding

Spatial Encoder. To obtain spatial priors, we encode the reference observations with a frozen spatial encoder $\text{Encoder}_{\text{spa}}$ pretrained on diverse static-scene NVS datasets (Szymanowicz et al., 2026).

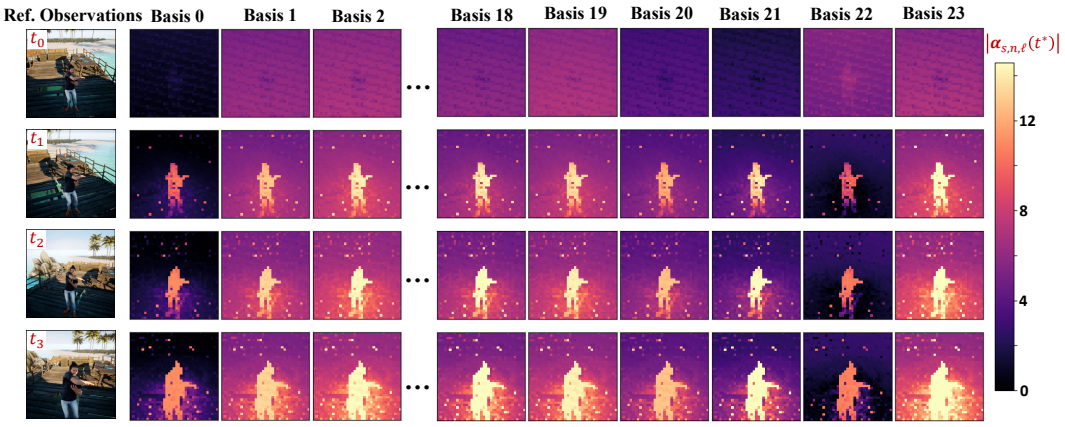

Figure 5: **Motion coefficient maps across reference observations.** The left column shows reference images at t_0–t_3; the remaining columns show coefficient magnitudes $|\alpha_{s,m}(t^*)|$ for selected motion bases at $t^* = t_0$. Responses are weak for the time-aligned reference at t_0 and strongest around the moving person in the other references, with varying strengths across bases.

Given the posed reference images, it extracts spatial tokens and intermediate features. Formally,

$$(\mathbf{z}_{\text{spa}}, \mathbf{z}_{\text{int}}) = \text{Encoder}_{\text{spa}}(\mathbf{I}, \Pi). \tag{3}$$

Motion Encoder. We extract intermediate features from layers [12, 16, 24] of the frozen spatial encoder, selected empirically (Appendix B). The multi-layer features are concatenated along the channel dimension, fused by an MLP into $\mathbf{z}_{\text{int}}$, then augmented with reference-ray and timestamp embeddings before being processed by a trainable Transformer-based motion encoder with alternating attention:

$$\mathbf{z}_{\text{mot}} = \text{Encoder}_{\text{mot}}(\mathbf{z}_{\text{int}} + \Phi_{\text{ray}}(\Pi) + \Phi_{\text{time}}(t)) \,. \tag{4}$$

Here Φ_{ray} embeds the reference ray at each token location, while Φ_{time} embeds the timestamp and broadcasts it across tokens from the same reference image. The motion encoder alternates attention within and across reference images to produce motion tokens $\mathbf{z}_{\text{mot}}$.

3.3 Factorized Temporal Adaptation and Rendering

We adapt the pretrained spatial tokens through a residual factorized into shared motion bases and local coefficients, following prior motion models (Akhter et al., 2008; Kratimenos et al., 2024). The bases describe scene-dependent directions of change; the coefficients select their contribution for each reference token at the target time t^*.

Motion Basis Decoder. The motion basis decoder learns a reference-dependent basis from the extracted motion tokens. We train a set of learnable queries $\mathbf{q}_{\text{basis}} \in \mathbb{R}^{K\times d}$ to aggregate motion tokens through cross-attention:

$$\mathbf{B} = \text{RowNorm}(\text{Decoder}_{\text{basis}}(\mathbf{q}_{\text{basis}}, \text{Concat}(\mathbf{z}_{\text{mot}}))) \,, \tag{5}$$

where $\mathbf{B} = [b_1, b_2, \cdots, b_K] \in \mathbb{R}^{K\times d}$ is specific to the unified reference set and independent of t^*.

Coefficient Decoder. For each reference s, we form target-time-conditioned queries from its spatial and motion tokens:

$$\mathbf{q}_{\text{coeff},s}(t^*) = \text{LayerNorm}(\mathbf{z}_{\text{spa},s} + \mathbf{z}_{\text{mot},s} + \Phi_{\text{coeff}}(\mathbf{r}_s, \mathbf{p}_s, t^*, t^* - t_s)) \,. \tag{6}$$

Here $\mathbf{r}_s$ and $\mathbf{p}_s$ are the reference ray and patch position at each token location. Φ_{coeff} combines them with the target time t^* and relative time $t^* - t_s$, which are shared across tokens of reference s, to produce an $N \times d$ embedding. Thus, $\mathbf{q}_{\text{coeff},s}(t^*) \in \mathbb{R}^{N\times d}$, where N is the number of tokens per reference. Using these queries and the motion tokens of reference s as keys and values, the coefficient decoder predicts

$$\mathbf{A}_s(t^*) = \text{Decoder}_{\text{coeff}}(\mathbf{q}_{\text{coeff},s}(t^*), \mathbf{z}_{\text{mot},s}) \in \mathbb{R}^{N\times K}, \tag{7}$$

with K coefficients for each reference token.

Motion Residual. The basis and coefficients combine into a motion residual, which updates the source spatial tokens:

$$\Delta\mathbf{z}_{\text{spa},s}(t^*) = g\, \mathbf{A}_s(t^*)\mathbf{B}, \tag{8}$$

$$\widetilde{\mathbf{z}}_{\text{spa},s}(t^*) = \mathbf{z}_{\text{spa},s} + \Delta\mathbf{z}_{\text{spa},s}(t^*), \tag{9}$$



Figure 6: **Qualitative comparison of spatial queries at a fixed target time.** Given the same reference observations and target time, we synthesize different target viewpoints (V_1 and V_2). Red boxes highlight enlarged regions for detailed comparison.

where g is a learnable scalar gain. At a fixed target time, the adapted tokens can be reused across target viewpoints; the target camera enters the subsequent render decoder.

Fig. 5 visualizes the coefficient magnitudes $|\alpha_{s,n,k}(t^*)|$ at their corresponding reference-image locations. The responses concentrate on the moving subject and remain weak over largely static regions, indicating that temporal adaptation is spatially localized. Moreover, the time-aligned reference exhibits substantially weaker responses, suggesting that smaller updates are required when the source state already matches the target time. The basis-dependent response patterns further show that local temporal changes are expressed through different combinations of the shared motion bases.

Render Decoder. We initialize the render decoder from pretrained LagerNVS (Szymanowicz et al., 2026), freeze its original weights, and fine-tune it with LoRA (Hu et al., 2022). For a fixed reference set, the spatial tokens, motion tokens, and motion basis matrix are computed once. Each target time t^* determines new coefficients and hence updated spatial tokens. Conditioned on the target camera Π^*, the render decoder directly synthesizes the target image from these updated tokens.

3.4 Losses

Reconstruction Loss. We optimize the motion branch and the renderer's LoRA parameters, while keeping the spatial encoder and the original renderer parameters frozen. Given a unified reference set $\mathcal{R}$, we sample eight target timestamps with eight target views at each timestamp, yielding $J = 64$ target observations. For each target, AnySpaceTime predicts $\hat{I}_j = \mathcal{M}_\theta(\mathcal{R}, t_j^*, \Pi_j^*)$, with I_j^* denoting the corresponding ground-truth target image. To optimize AnySpaceTime, we use a combined loss function consisting of the MSE Loss and LPIPS loss (Zhang et al., 2018) as

$$\mathcal{L}_{\text{recon}} = \frac{1}{J}\sum_{j=1}^{J}\left[\lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}}(\hat{I}_j, I_j^*) + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}}(\hat{I}_j, I_j^*)\right]. \tag{10}$$

where $\lambda(\cdot)$ are the coefficients.

Orthogonality Loss. To discourage redundant motion-basis directions, we regularize the row-normalized basis matrix $\mathbf{B} \in \mathbb{R}^{K\times d}$ defined in Sec. 3.3, where K is the number of motion bases and d is their feature dimension:

$$\mathcal{L}_{\text{ortho}}(\mathbf{B}) = \frac{2}{K(K-1)}\sum_{k=1}^{K-1}\sum_{k'=k+1}^{K}\left(\mathbf{b}_k^\top \mathbf{b}_{k'}\right)^2, \qquad K > 1. \tag{11}$$



Figure 7: **Qualitative comparison of temporal queries at a fixed target viewpoint.** Given the same reference observations and target viewpoint, we synthesize different target times (T_1 and T_2). Red boxes highlight enlarged regions for detailed comparison.

This loss penalizes correlations between distinct motion bases, encouraging them to capture complementary directions of temporal variation.

Total Loss. The overall training objective combines the reconstruction loss with the motion-basis regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{ortho}} \mathcal{L}_{\text{ortho}}(\mathbf{B}), \tag{12}$$

where λ_{ortho} controls the strength of the orthogonality regularization.

4 Experiments

AnySpaceTime is trained under the unified reference formulation of different observation patterns defined in Sec. 3.1. We evaluate a single model across three representative settings: monocular-video observations (Fig. 3(b)), synchronous and asynchronous multi-view observations (Fig. 3(c) and (d)), and static multi-view observations (Fig. 3(a)). We further conduct ablation studies to assess the contribution of each component of AnySpaceTime.

4.1 Experimental Setup

Training datasets. Following the unified reference formulation introduced in Sec. 3.1, we train our method on ARKitScenes (Baruch et al., 2021), SynCamVideo (Bai et al., 2025), Ego-Exo4D (Grauman et al., 2024), and the table- and room-scale subsets of ACE-Data-0 (Cao et al., 2026).

Evaluation protocol. The same model is applied to all settings without setting-specific fine-tuning, allowing us to evaluate its ability to handle different spatial and temporal sampling patterns within a unified framework. On all quantitative tasks we use standard metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Image Similarity (SSIM (Wang et al., 2004)), and Perceptual Similarity (LPIPS (Zhang et al., 2018)).

Implementation details. We initialize AnySpaceTime from LagerNVS (Szymanowicz et al., 2026) and freeze the spatial encoder. The motion branch and rank-8 renderer LoRA (Hu et al., 2022) are trained for 16,000 iterations at 512×512 resolution with eight reference images per example and 24 motion bases. Temporal adaptation uses target RGB supervision and basis orthogonality regularization. More details are provided in Appendix D.

Table 1: **Quantitative comparison for Space-Time NVS with monocular-video observations.** Results on three dynamic-scene benchmarks, Neural 3D Video (Li et al., 2022), Immersive (Broxton et al., 2020), and PKU-DyMVHumans (Zheng et al., 2024), using the monocular-video observation pattern illustrated in Fig. 3(b). ★ denotes C4G without VDM, and ☆ denotes NeoVerse without refinement. The best and second-best results are highlighted in bold and underlined, respectively.

	Neural 3D Video			Immersive			PKU-DyMVHumans		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
4DGT (Xu et al., 2025)	12.30	0.415	0.549	11.21	0.348	0.623	17.32	<u>0.715</u>	0.373
C4G★ (Kim et al., 2026)	14.55	0.465	0.554	<u>14.09</u>	<u>0.505</u>	0.772	16.72	0.531	0.406
MoVieS (Lin et al., 2026a)	<u>18.12</u>	<u>0.652</u>	<u>0.211</u>	13.46	0.435	<u>0.439</u>	<u>20.55</u>	0.571	<u>0.210</u>
NeoVerse☆ (Yang et al., 2026)	12.40	0.360	0.522	10.49	0.233	0.771	16.17	0.616	0.461
NoPo4D (Balice et al., 2026)	13.20	0.508	0.339	7.51	0.400	0.658	10.41	0.448	0.528
AnySpaceTime	**22.04**	**0.707**	**0.155**	**20.43**	**0.622**	**0.311**	**22.93**	**0.842**	**0.139**

Table 2: **Quantitative comparison for Space-Time NVS with synchronous and asynchronous multi-view observations.** Results on Neural 3D Video (Li et al., 2022) using the synchronous and asynchronous multi-view observation patterns illustrated in Fig. 3(c) and (d), respectively. The best and second-best results are highlighted in bold and underlined, respectively.

	Synchronous Multi-view			Asynchronous Multi-view		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
4DGT (Xu et al., 2025)	18.15	0.579	0.393	18.11	0.586	0.383
MoVieS (Lin et al., 2026a)	<u>21.81</u>	**0.740**	<u>0.151</u>	<u>21.74</u>	**0.742**	<u>0.153</u>
NoPo4D (Balice et al., 2026)	17.39	0.540	0.240	17.10	0.541	0.241
AnySpaceTime	**22.87**	<u>0.713</u>	**0.117**	**22.94**	<u>0.714</u>	**0.116**

4.2 Monocular Dynamic Novel View Synthesis

For the monocular dynamic sequence setting in Fig. 3(b), we use eight frames from a monocular video of a dynamic scene as reference observations. We then evaluate target queries at both observed reference times and unobserved intermediate times within the reference interval, with detailed protocols provided in Appendix D.2. Table 1 reports quantitative results on Neural 3D Video (Li et al., 2022), Immersive (Broxton et al., 2020), and PKU-DyMVHumans (Zheng et al., 2024). AnySpaceTime achieves the best performance across all metrics and datasets, with PSNR improvements of 3.93, 6.34, and 2.35 dB over the strongest baseline, respectively. Qualitative comparisons are shown in Figs. 6 and 7, which evaluate viewpoint variation at a fixed target time and temporal variation at a fixed target viewpoint, respectively.

4.3 Multi-View Spacetime Novel View Synthesis

We further evaluate AnySpaceTime on Neural 3D Video (Li et al., 2022) under the synchronous and asynchronous multi-view settings illustrated in Fig. 3(c) and (d). The synchronous references share timestamps across cameras, while the asynchronous references are temporally misaligned; both settings use the same 64 target queries. As shown in Table 2, AnySpaceTime outperforms the strongest baseline by 1.06 and 1.20 dB in PSNR, respectively, while maintaining nearly unchanged performance across the two settings.

4.4 Static Novel-View Synthesis

We further evaluate AnySpaceTime on static multi-view NVS using the observation pattern in Fig. 3(a), examining whether the added temporal modeling preserves the spatial capability inherited from LagerNVS (Szymanowicz et al., 2026). As shown in Table 3, AnySpaceTime remains close to its backbone across three datasets. Notably, the full temporal branch remains active during evaluation, indicating that temporal modeling can be introduced without substantially compromising the pretrained static NVS capability.

4.5 Ablation Studies and Analysis

We analyze the contribution of the main components of AnySpaceTime, as shown in Table 4.

Table 3: **Quantitative comparison for Space-Time NVS with static multi-view observations.** Results on the static-scene benchmarks RealEstate10K (Zhou et al., 2018), DL3DV (Ling et al., 2024), and ACID (Liu et al., 2021), using the static multi-view observation pattern illustrated in Fig. 3(a). LVSM (Jin et al., 2025) and LagerNVS (Szymanowicz et al., 2026) are trained specifically for static-scene NVS and do not model temporal scene dynamics. The best and second-best results are highlighted in bold and underlined, respectively.

	RealEstate10K			DL3DV			ACID		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
LVSM (Jin et al., 2025)	**30.81**	**0.910**	**0.062**	15.86	0.496	0.473	**31.69**	**0.878**	**0.089**
LagerNVS (Szymanowicz et al., 2026)	<u>29.85</u>	<u>0.899</u>	0.075	**19.65**	**0.601**	<u>0.321</u>	<u>30.89</u>	<u>0.860</u>	0.122
4DGT (Xu et al., 2025)	14.09	0.590	0.371	15.47	0.568	0.388	8.68	0.218	0.688
MoVieS (Lin et al., 2026a)	24.19	0.833	0.126	14.41	0.427	0.431	24.27	0.774	0.180
NoPo4D (Balice et al., 2026)	12.10	0.541	0.466	9.85	0.517	0.598	12.29	0.400	0.470
AnySpaceTime	29.51	0.894	<u>0.068</u>	<u>19.18</u>	<u>0.581</u>	**0.313**	30.59	0.856	<u>0.102</u>

Table 4: **Component ablation of AnySpaceTime.**

Variant	PSNR↑	SSIM↑	LPIPS↓
w/o Unified Reference Set	26.77	0.869	0.117
w/o Motion Basis	26.65	0.862	0.121
w/o Renderer LoRA	26.83	0.873	0.115
w/o $\mathcal{L}_{\text{ortho}}$	26.88	0.875	0.113
Ours	**27.26**	**0.889**	**0.109**

Table 5: **Effect of motion-basis capacity.** We vary the number of motion bases K while keeping all other model and training settings fixed.

Setting	PSNR↑	SSIM↑	LPIPS↓
$K = 8$	26.83	0.876	0.115
$K = 16$	26.71	0.868	0.119
$K = 24$ (default)	**27.26**	**0.889**	**0.109**
$K = 32$	27.20	0.887	0.110

Unified Reference Set. The Unified Reference Set (Sec. 3.1) provides a common representation for the four observation patterns shown in Fig. 3. To evaluate its benefit during training, we train a variant using only the conventional monocular-video observation pattern. The resulting performance drop relative to the full model demonstrates that training with the Unified Reference Set improves model performance.

Motion Basis. Replacing the shared motion-basis representation (Sec. 3.3) with direct token-residual prediction causes the largest performance drop among the component ablations. This result demonstrates the importance of modeling temporal updates in a shared low-dimensional basis space rather than predicting unconstrained residuals directly. To examine the capacity of this representation, we vary the number of bases K in Table 5. Performance peaks at $K = 24$, with no further gain when K increases to 32. We therefore use $K = 24$ throughout our experiments.

Renderer LoRA Adaptation. Removing the renderer LoRA causes a performance drop, indicating that lightweight adaptation helps the pretrained renderer decode temporally updated features. The LoRA update enables it to accommodate these features while retaining its learned spatial prior.

Motion-basis Orthogonality Loss. Removing the orthogonality loss leads to a performance drop, indicating that encouraging the learned motion bases to capture complementary directions improves the shared motion representation.

5 Conclusion

We introduced Space-Time Novel View Synthesis, a unified formulation for rendering target times and viewpoints from posed and timestamped observations with diverse sampling patterns. Based on this formulation, AnySpaceTime preserves a pretrained NVS spatial representation and models dynamics through structured latent adaptation with context-conditioned motion bases and source-local coefficients. Across static, monocular dynamic, and synchronized/asynchronous multi-view settings, a single model achieves high-fidelity synthesis while largely retaining the pretrained static NVS capability. These results suggest that dynamic scene modeling can be introduced directly in a renderable latent space, without requiring explicit dynamic 3D/4D reconstruction or auxiliary motion supervision.

AnySpaceTime currently assumes known camera parameters and focuses primarily on interpolation within the observed temporal range. Its performance may degrade when the input observations provide insufficient evidence for heavily occluded content, rapid non-rigid motion, or large temporal extrapolation. Future work could relax the pose requirement, explicitly model uncertainty under ambiguous observations, and extend the framework toward longer and more complex scene dynamics.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and code debugging. But all technical content, experimental design, results, and conclusions were independently verified by the authors, who take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training procedure, data processing, and evaluation protocol in the main paper and Appendix. Additional implementation details, hyperparameters, and qualitative results are also included to facilitate reproduction of our results.

ETHICS STATEMENT

This work focuses on dynamic novel view synthesis and does not involve human subjects or the collection of personally identifiable information. We are not aware of any specific ethical concerns beyond those generally associated with generative visual models.

REFERENCES

Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/dc82d632c9fcecb0778afbc7924494a6-Paper.pdf.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. SynCamMaster: Synchronizing multi-camera video generation from diverse viewpoints. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.07760.

Matteo Balice, Yanik Kunzi, Chenyangguang Zhang, Matteo Matteucci, Marc Pollefeys, and Sungwhan Hong. No pose, no problem in 4D: Feed-forward dynamic gaussians from unposed multi-view videos, 2026. URL https://arxiv.org/abs/2605.22190.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics*, 39(4):86:1–86:15, 2020. doi: 10.1145/3386569.3392485.

Yukang Cao, Haozhe Xie, Beichen Wen, Runmao Yao, Yinghao Liu, Yue Huang, Zhichao Liao, Yunxiang Wang, Haiheng Liu, Xingshun Tian, Dawei Su, Long Zhuo, Dacheng Tao, Xiaogang Wang, Liang Pan, and Ziwei Liu. ACE-Data-0: Human-centric ambient capture as embodied data engine, 2026. URL https://arxiv.org/abs/2607.28625.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2312.12337.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-View Images. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2403.14627.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.18259.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: A Self-supervised Large View Synthesis Model. *arXiv preprint arXiv:2505.00702*, 2025. URL https://arxiv.org/abs/2505.00702.

Yanqin Jiang, Tengfei Wang, Zhengwei Wang, Chenjie Cao, Junta Wu, Wenhan Luo, Weiming Hu, Jin Gao, and Chunchao Guo. OmniX: Any-view and Any-time 4D Reconstruction via Feed-forward Trajectory Fields. *arXiv preprint arXiv:2607.10840*, 2026. URL https://arxiv.org/abs/2607.10840.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.17242.

Mungyeom Kim, Minkyeong Jeon, Honggyu An, Jaewoo Jung, Hyuna Ko, Jisang Han, Hyeonseo Yu, Donghwan Shin, Sunghwan Hong, Takuya Narihira, Kazumi Fukuda, Yuki Mitsufuji, and Seungryong Kim. Learning global motion with compact gaussians for feed-forward 4D reconstruction, 2026. URL https://arxiv.org/abs/2605.31595.

Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. DynMF: Neural motion factorization for real-time dynamic view synthesis with 3D gaussian splatting. In *European Conference on Computer Vision*, 2024.

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://neural-3d-video.github.io/.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, and Jiahui Huang. Feed-Forward Bullet-Time Reconstruction of Dynamic Scenes from Monocular Videos. *arXiv preprint arXiv:2412.03526*, 2024. URL https://arxiv.org/abs/2412.03526.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. MoVieS: Motion-aware 4D dynamic view synthesis in one second. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. URL https://arxiv.org/abs/2507.10065.

Chieh Hubert Lin, Zhaoyang Lv, Songyin Wu, Zhen Xu, Thu Nguyen-Phuoc, Hung-Yu Tseng, Julian Straub, Numair Khan, Lei Xiao, Ming-Hsuan Yang, Yuheng Ren, Richard Newcombe, Zhao Dong, and Zhengqin Li. DGS-LRM: Real-Time Deformable 3D Gaussian Reconstruction From Monocular Videos. *arXiv preprint arXiv:2506.09997*, 2025. URL https://arxiv.org/abs/2506.09997.

Shing Ho J. Lin, Wenzhao Zheng, Dong Zhuo, Yuqi Wu, Jie Zhou, and Jiwen Lu. SM4RT: Learning Structured Motion Geometry for 4D Reconstruction. *arXiv preprint arXiv:2607.22534*, 2026b. URL https://arxiv.org/abs/2607.22534.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024. URL https://dl3dv-10k.github.io/DL3DV-10K/.

Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. URL https://infinite-nature.github.io/.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4RC: 4D Reconstruction via Conditional Querying Anytime and Anywhere. In *International Conference on Machine Learning*, 2026. URL https://yihangluo.com/projects/4RC/.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.

Chris Rockwell, Joseph Tung, Tsung-Yi Lin, Ming-Yu Liu, David F. Fouhey, and Chen-Hsuan Lin. Dynamic camera poses and where to find them. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Qiuhong Shen, Xuanyu Yi, Mingbao Lin, Hanwang Zhang, Shuicheng Yan, and Xinchao Wang. Seeing World Dynamics in a Nutshell. *arXiv preprint arXiv:2502.03465*, 2025. URL https://arxiv.org/abs/2502.03465.

Chen Shi, Shaoshuai Shi, Xiaoyang Lyu, Chunyang Liu, Kehua Sheng, Bo Zhang, and Li Jiang. Unisplat: Unified spatio-temporal fusion via 3d latent scaffolds for dynamic driving scene reconstruction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Ng2VDbKD4r.

Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. LagerNVS: Latent geometry for fully neural real-time novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.20176.

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*, 2025. URL https://arxiv.org/abs/2407.13764.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. URL https://ece.uwaterloo.ca/~z70wang/research/ssim/.

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*, 2024. URL https://arxiv.org/abs/2310.08528.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4DGT: Learning a 4D gaussian transformer using real-world monocular videos. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.08015.

Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Maximilian Igl, Apoorva Sharma, Peter Karkus, Danfei Xu, Boris Ivanovic, Yue Wang, and Marco Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2025.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. NeoVerse: Enhancing 4D world model with in-the-wild monocular videos, 2026. URL https://arxiv.org/abs/2601.00393.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently Reconstructing Dynamic Scenes One D4RT at a Time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2512.08924.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Xiaoyun Zheng, Liwei Liao, Xufeng Li, Jianbo Jiao, Rongjie Wang, Feng Gao, Shiqi Wang, and Ronggang Wang. PKU-DyMVHumans: A multi-view video benchmark for high-fidelity dynamic human modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37, 2018. URL https://research.google/pubs/stereo-magnification-learning-view-synthesis-using-multiplane-images/.

A Spatial Features under Dynamic Observations

LagerNVS (Szymanowicz et al., 2026) is trained for static novel-view synthesis, where all reference observations describe a common scene state. Dynamic observations violate this assumption because appearance changes caused by scene motion are mixed with changes caused by viewpoint. We use a series of rendering interventions to examine what remains accessible in the frozen representation. The analysis addresses two questions: whether the pretrained representation still provides a useful spatial prior, and whether it has already combined dynamic observations into a coherent target-time state. The results support a consistent interpretation: static content remains stable across reference observations, whereas dynamic content retains source-associated states that remain locally renderable. This behavior motivates the source-anchored temporal adaptation introduced in Sec. 3.3 of the main paper.

A.1 Frozen Rendering with Mixed-Time Context

We first apply frozen LagerNVS directly to reference observations captured at different viewpoints and times, without providing timestamps or performing temporal adaptation. As shown in Fig. 2 of the main paper, the static background is rendered consistently at the target views, while moving people are blurred or contain mixtures of incompatible poses. The pretrained representation therefore continues to support camera-conditioned rendering of static scene structure, but direct aggregation does not select a coherent state for dynamic content.

Viewpoint variation is not the sole cause of this behavior. In Fig. 8, all reference observations use a fixed camera while the scene changes over time. Frozen LagerNVS again preserves the static background but blends the moving subject. Removing viewpoint variation thus does not resolve the temporal inconsistency: the representation contains the observed dynamic states, but the static model has no target-time signal with which to reconcile them.

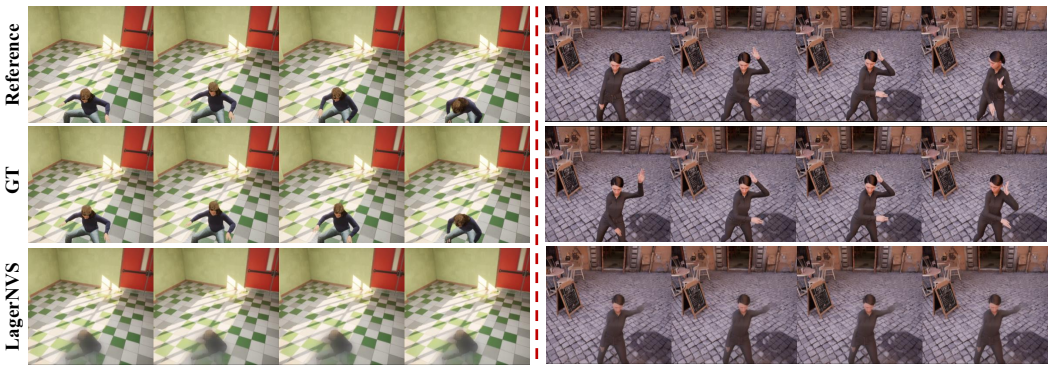

Figure 8: **Fixed-camera control.** The context observations share the same camera and differ only in scene time. Frozen LagerNVS preserves the static regions but averages the moving subject, showing that the failure persists after viewpoint variation is removed.

A.2 Source-Aligned Queries

We next ask whether the blurred dynamic content reflects information loss or the coexistence of source-dependent states. Starting from the same mixed-time context, we set the target camera equal to one of the reference cameras, $\Pi^* = \Pi_s$. As shown in Fig. 9, the dynamic region becomes substantially sharper and follows the state visible in the camera-matched source. Repeating the intervention with different reference cameras produces the corresponding source-associated state. The frozen representation has therefore not simply discarded the dynamic content or collapsed all observations into a single average. Instead, source-specific appearance remains accessible to the renderer through the source-anchored spatial tokens.

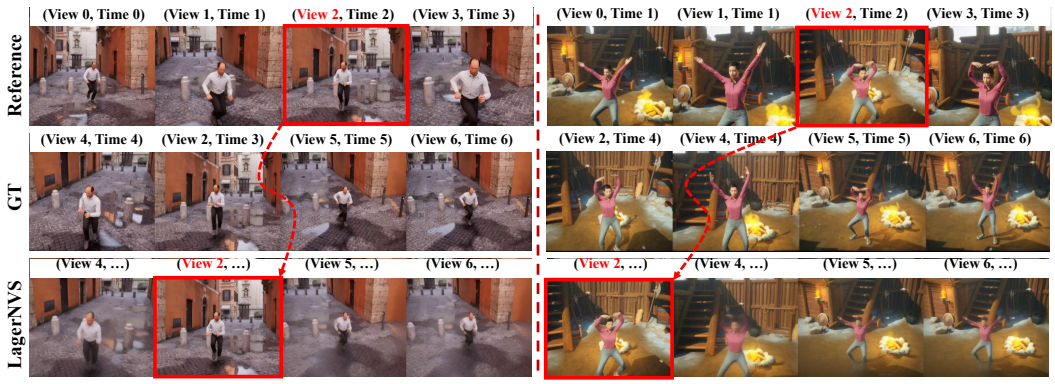

Figure 9: **Source-aligned camera queries.** The highlighted context observation supplies a camera used again as the target camera. Predictions at that camera become sharp and reproduce the source-associated dynamic state, while queries at other viewpoints continue to mix states from different times.

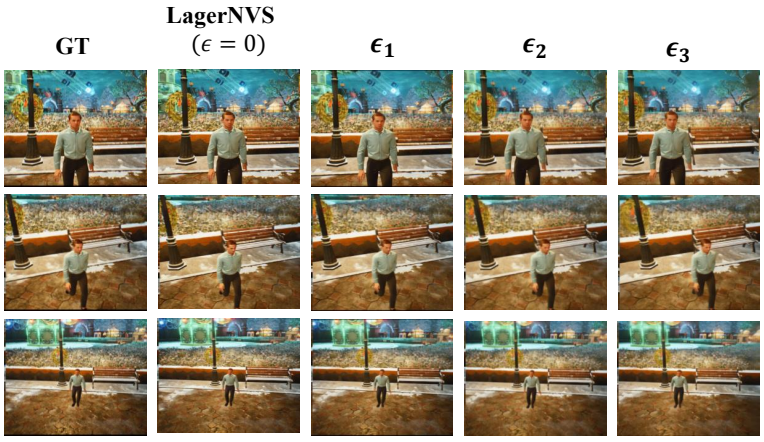

Figure 10: **Local perturbations around a source-aligned camera.** Starting from the matched camera ($\epsilon = 0$), we perturb the target pose with increasing magnitude. The renderer changes the viewpoint while retaining the source-associated dynamic appearance, indicating that the effect extends beyond an exact camera match.

This effect is not restricted to exact camera matching. We perturb the target pose around the matched source camera,

$$\widetilde{\Pi}_s = \Pi_s \oplus \Delta\Pi, \qquad \widetilde{\Pi}_s \neq \Pi_s, \tag{13}$$

and progressively increase the perturbation magnitude. Fig. 10 shows that nearby views preserve the source-associated appearance while changing the rendered viewpoint. This rules out an explanation based only on exact image retrieval and indicates that the source state remains locally renderable around its observed camera.

A.3 Source-Feature Masking

Finally, we test whether the camera-matched source tokens are sufficient for this source-aligned behavior. For a query with $\Pi^* = \Pi_s$, we retain the spatial tokens $\mathbf{z}_{\mathrm{spa},s}$ from the matched observation and mask the tokens from the remaining observations before rendering. Fig. 11 shows that the masked prediction remains close to the full-context prediction in both static and dynamic regions. The corresponding source tokens thus retain most of the information used for source-aligned reconstruction. This intervention does not imply that other observations are unnecessary for novel viewpoints or occluded regions; it isolates where the visible source-associated state is stored.

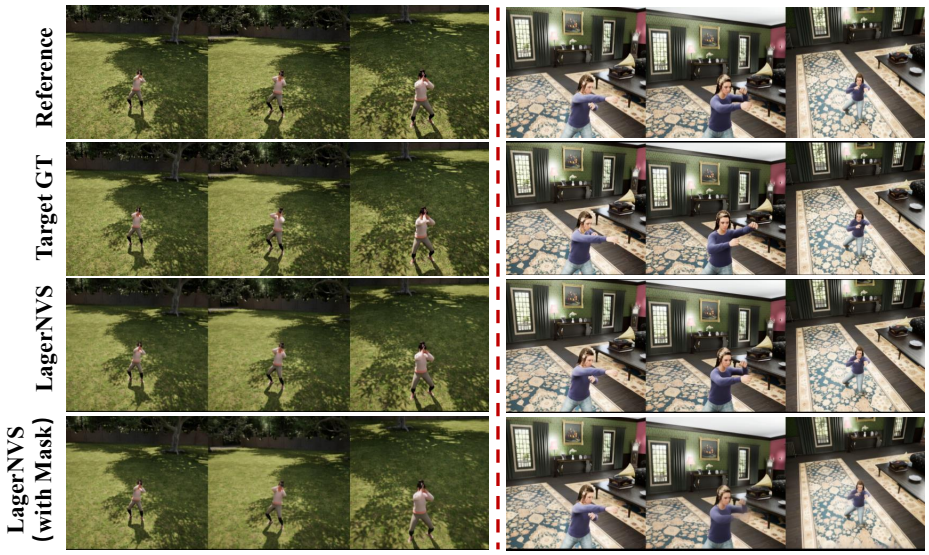

Figure 11: **Source-feature masking.** For each source-aligned query, the final row masks spatial tokens from all observations except the camera-matched source. The prediction remains close to the full-context LagerNVS output, including in the dynamic region, showing that the matched source tokens retain the source-associated state.

A.4 Implications for Temporal Adaptation

Taken together, these probes distinguish a useful spatial prior from a time-consistent dynamic representation. The frozen encoder preserves static scene structure across observations and retains renderable, source-associated information about dynamic regions. What is missing is a mechanism that transforms these distinct source-time states into the state requested at t^*. AnySpaceTime therefore keeps the source-anchored spatial tokens and predicts a residual for each source observation, as defined in Eq. 8. The target time controls this latent update, after which the target camera controls rendering. This separation uses the spatial structure already present in the pretrained model while learning the temporal reconciliation that the frozen static model lacks.

Importantly, the parameterization does *not* impose a hard zero-update constraint when the source and target times coincide. Static scenes and time-aligned references are handled by the same model, and small temporal updates are an empirical behavior rather than a structural guarantee. In this paper, the "zero-motion case" refers to the task condition in which all observations share a common scene state, rather than to a parameterization constraint that enforces an exactly zero latent residual.

A.5 Coefficient Magnitudes versus Actual Latent Updates

For a spatial token with coefficient row vector $\mathbf{a}_{s,n}(t^*) \in \mathbb{R}^K$, the modeled residual is

$$\Delta \mathbf{z}_{\mathrm{spa},s,n}(t^*) = g\, \mathbf{a}_{s,n}(t^*)\mathbf{B}, \tag{14}$$

where $\mathbf{B} \in \mathbb{R}^{K\times d}$ is the motion-basis matrix. Its squared norm is

$$\|\Delta \mathbf{z}_{\mathrm{spa},s,n}(t^*)\|_2^2 = g^2\, \mathbf{a}_{s,n}(t^*)\mathbf{B}\mathbf{B}^\top \mathbf{a}_{s,n}(t^*)^\top. \tag{15}$$

Therefore, the magnitude of any single coefficient is *not* by itself equal to the magnitude of the resulting latent update. The coefficient maps in Fig. 5 of the main paper should consequently be interpreted as qualitative evidence of spatially varying basis activation, rather than as a direct measurement of temporal-update strength. Accordingly, Fig. 5 visualizes where and how different basis directions are activated, but does not by itself establish the spatial magnitude of the resulting temporal adaptation.

A.6 Notation Clarification

Throughout the supplement, K denotes the number of motion bases. We use $\mathbf{z}_{\mathrm{spa}}$ for spatial tokens, $\mathbf{A}_s(t^*) \in \mathbb{R}^{N\times K}$ for the coefficient matrix associated with reference observation s, and $\mathbf{B} \in \mathbb{R}^{K\times d}$ for the motion-basis matrix. The default configuration uses $K = 24$. We denote the training loss weights by λ_{MSE}, λ_{LPIPS}, and λ_{ortho}.

B Selecting Intermediate Features for Motion Modeling

The motion branch reads intermediate activations from the frozen spatial encoder. We therefore examine whether these features retain information about scene dynamics and which encoder depths provide useful motion cues. Experiments are conducted on Dynamic Replica using the pretrained 24-layer LagerNVS encoder, which remains frozen throughout the analysis. Let

$$Z_{s,n}^{(\ell)} \in \mathbb{R}^D$$

denote the intermediate token at encoder layer ℓ and patch-grid location n for observation s, where $D = 2048$.

Raw feature variation. We first test whether temporal changes in the frozen features correlate with dynamic image regions. For two temporally separated observations s and s', we normalize the token features along the channel dimension,

$$\widetilde{Z}_{s,n}^{(\ell)} = \frac{Z_{s,n}^{(\ell)}}{\|Z_{s,n}^{(\ell)}\|_2}, \tag{16}$$

and measure the variation magnitude

$$d_{s,s',n}^{(\ell)} = \left\| \widetilde{Z}_{s',n}^{(\ell)} - \widetilde{Z}_{s,n}^{(\ell)} \right\|_2 . \tag{17}$$

We evaluate the alignment of $d^{(\ell)}$ with dynamic regions using the dynamic/static ratio, average precision (AP), correlation with the dynamic mask, and Top-20% overlap. Across temporal strides of 5, 8, and 10, dynamic regions consistently exhibit larger feature changes than static regions, with dynamic/static ratios of approximately 1.82–2.00. Direct localization using the scalar variation magnitude is strongest in shallow layers: L2 achieves the highest AP (0.664), while L1 gives the highest correlation (0.575) and Top-20% overlap (0.658). This ordering is consistent across the three temporal strides.

Linear motion probes. The scalar magnitude $d^{(\ell)}$ captures how strongly a token changes, but discards the channel-wise structure of that change. We therefore retain

$$v_{s,s',n}^{(\ell)} = \left| \widetilde{Z}_{s',n}^{(\ell)} - \widetilde{Z}_{s,n}^{(\ell)} \right| \in \mathbb{R}^D \tag{18}$$

and train lightweight linear probes while keeping the encoder fixed. We consider two complementary targets. The **dynamic-region probe** predicts whether a patch belongs to a dynamic region, while the **motion-magnitude probe** regresses the displacement magnitude of visible Dynamic Replica trajectories. For multi-layer probes, features from several encoder depths are concatenated along the channel dimension. The probe consists of a normalization layer followed by a linear prediction head.

Table 6 reports the main results on the larger 48/16 sequence split. In contrast to raw variation magnitude, supervised readout favors intermediate layers. L12 gives the highest single-layer AP for dynamic-region classification (0.847), while L16 gives the lowest single-layer trajectory RMSE (0.170) and highest correlation (0.557). Combining multiple depths provides a further improvement: L8+12+16 reaches an AP of 0.885, while L12+16+24 improves trajectory correlation to 0.704 with an RMSE of 0.146.

The two analyses reveal complementary behavior across encoder depth. Shallow layers are most sensitive to direct local appearance changes, whereas motion-related information becomes more linearly accessible in intermediate layers. Multi-layer probes further indicate that different depths

Table 6: Linear probing of frozen intermediate features. AP evaluates dynamic-region prediction; RMSE and Pearson correlation evaluate motion-magnitude regression. The encoder is frozen for all probes.

Encoder features	Dynamic AP ↑	Motion RMSE ↓	Motion Corr. ↑
L1	0.806	0.193	0.394
L4	0.809	0.189	0.403
L8	0.812	0.181	0.477
L12	**0.847**	0.173	0.538
L14	0.843	0.172	0.550
L16	0.829	**0.170**	**0.557**
L20	0.791	0.173	0.536
L24	0.793	0.175	0.532
L12+16	0.880	0.151	0.671
L8+12+16	**0.885**	0.150	0.677
L12+16+24	0.882	**0.146**	**0.704**

contain complementary cues; in particular, L24 is not the strongest layer in isolation, but improves motion-magnitude prediction when combined with L12 and L16.

These observations motivate using intermediate frozen encoder features as inputs to the motion branch rather than relying only on the final encoder representation. We use the probing results as evidence that such intermediate features contain accessible motion cues and to guide feature selection. Importantly, the probes measure representation quality rather than downstream rendering quality; they do not by themselves establish that a particular layer combination is optimal for final novel-view synthesis.

C Empirical Structure and Rank of Latent Temporal Updates

Matched token residuals. We examine temporal differences in the frozen LagerNVS token space before imposing the factorization used by AnySpaceTime. The selected camera poses are exactly fixed across the two times for all ten cases: $\mathrm{Pose}(t_i, v_j) = \mathrm{Pose}(t_k, v_j)$. Each view therefore receives the same camera token at both times. For object patches, pixel-level Face-ID flow maps the patch centroid to the same object face at the other time; for background patches, mutual depth and camera reprojection with an occlusion and depth check establishes correspondence. Boundary and low-purity object patches are excluded. For matched object tokens, the measured residual is

$$\mathbf{r}_{j,p} = \mathbf{f}(t_k, v_j, p') - \mathbf{f}(t_i, v_j, p) \in \mathbb{R}^{768}, \tag{19}$$

where p' is the Face-ID-flow match of patch p. Background residuals use the verified same-pixel correspondence permitted by the fixed camera poses.

We use two encoding protocols. Type 1 jointly encodes all eight observations $[A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4]$ and compares tokens at fixed view IDs across the two times. Type 2 separately encodes $[A_1, A_2, A_3, A_4]$ and $[A_1, B_2, B_3, B_4]$, then compares the corresponding tokens from views 2–4. The second protocol also exposes a context effect: although A_1 is identical in both inputs, changing its companion observations changes its output tokens. Its median background and object angular changes are 0.021 and 0.059 radians, respectively. Thus the measured residuals include effects of joint attention as well as scene evolution; they are not a direct ground truth for a pure physical motion field.

Energy concentration. For each protocol and case, we form a residual matrix from matched patches and compute its singular values $\sigma_1 \geq \sigma_2 \geq \cdots$. We report cumulative energy $E(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_i \sigma_i^2$ and $R_{90} = \min\{k : E(k) \geq 0.9\}$. The SVD is computed separately for object and background regions within each protocol-case, with at most 512 sampled residuals per group; Table 7 aggregates the ten case-level results by median. It is not an SVD over all residuals pooled across scenes.

Across both protocols, approximately 32 SVD directions capture 90% of the object-residual energy in a 768-dimensional token space. The object R_{90} values range from 18 to 37 across the 20 protocol-

Table 7: Matched-token residual spectra in the 10-case fixed-camera Syn4D pilot. Entries are medians across cases. Coordinate R_{90} sorts the original feature coordinates by residual energy, rather than using SVD directions.

Protocol	Region	SVD $R_{90} \downarrow$	$E(16) \uparrow$	$E(32) \uparrow$	Coordinate $R_{90} \downarrow$
Type 1: joint encoding	Background	17.0	89.6%	–	549.5
	Object	32.5	81.2%	90.1%	577.5
Type 2: reference split	Background	19.5	88.2%	–	546.5
	Object	32.0	81.5%	90.4%	569.5

case results, and every result has $R_{90} \leq 40$. By contrast, ranking the original feature coordinates by energy requires approximately 570 coordinates to reach the same threshold. The temporal residuals are therefore concentrated in a low-dimensional *directional* subspace within each case, rather than in a small fixed subset of feature channels. This provides empirical motivation for representing updates as weighted combinations of learned basis directions. The higher object R_{90} and lower $E(16)$ relative to background also indicate that dynamic regions require more directions than background changes.

The pilot supports a case-local, fixed-camera statement. Each case contains one temporal pair, so the spectra do not establish that one basis spans multiple target times or that a basis is shared across scenes. The default AnySpaceTime model uses $K = 24$ motion bases, but the pilot reports $E(16)$ and $E(32)$ rather than $E(24)$; it therefore does not establish the energy captured by 24 measured-token directions. The basis count is evaluated through downstream rendering ablations. The measured tokens are 768-dimensional; the following model analysis uses the spatial-token dimension d and does not assume that the two token spaces are identical.

Rank of the modeled update. For a fixed reference set, stack the coefficient matrices for all S observations at target time t^* as

$$\mathbf{A}(t^*) = \begin{bmatrix} \mathbf{A}_1(t^*) \\ \vdots \\ \mathbf{A}_S(t^*) \end{bmatrix} \in \mathbb{R}^{SN \times K}.$$

The context-dependent basis matrix $\mathbf{B} \in \mathbb{R}^{K \times d}$ is shared across these observations. Likewise, let $\Delta\mathbf{Z}_{\text{spa}}(t^*) \in \mathbb{R}^{SN \times d}$ denote the spatial-token residuals stacked over all reference observations. Equation 8 then gives

$$\Delta\mathbf{Z}_{\text{spa}}(t^*) = g\, \mathbf{A}(t^*)\mathbf{B}, \tag{20}$$

and therefore

$$\operatorname{rank}(\Delta\mathbf{Z}_{\text{spa}}(t^*)) \leq \min\{\operatorname{rank}(\mathbf{A}(t^*)), \operatorname{rank}(\mathbf{B})\} \leq K. \tag{21}$$

The same $\mathbf{B}$ is reused for every target time queried from this reference set. Stacking the modeled residual matrices for several target times along the token dimension therefore also yields rank at most K. This bound describes the parameterization; the independently measured spectra above provide empirical motivation for using a compact basis.

D Implementation Details and Additional Evaluations

D.1 Architecture and Optimization

AnySpaceTime initializes its spatial encoder and renderer from LagerNVS. The spatial encoder remains frozen. Selected intermediate encoder tokens are augmented with source-ray and timestamp embeddings, then processed by the motion encoder, which alternates attention within and across reference observations. The motion-basis decoder aggregates these motion tokens with $K = 24$ learned queries and row-normalizes the resulting bases. For each target time, the coefficient decoder combines each source token with its ray, patch position, target time, and source-to-target time offset. Its coefficients form the residual in Eq. 8; the adapted spatial tokens are then passed to the LoRA-adapted renderer with the target camera. Spatial tokens, motion tokens, and bases are computed once per reference set and reused across target queries, while target-time coefficients and rendering are query dependent.

Target-sampling clarification. Each training example uses eight target timestamps: four coincide with reference timestamps and four do not. Eight target views are supervised at each timestamp, yielding 64 target images in total. The 16–48 range reported for reference sampling denotes the temporal span, in frames, of the sampled reference window; it does not denote the number of target observations.

The motion branch and renderer LoRA parameters are trained jointly with AdamW for 16,000 iterations using 512×512 images and eight reference observations per example. The global batch size is 16. Training examples are sampled from five reference-observation patterns: fixed-view temporal sequences (20%), fixed-time multi-view captures (10%), monocular moving-camera trajectories with coupled changes in time and viewpoint (50%), synchronized multi-view sequences (10%), and asynchronous multi-view observations (10%). The renderer LoRA rank is eight. Training uses target RGB supervision with MSE, LPIPS, and basis orthogonality regularization as specified in the main paper; no depth, optical flow, scene flow, or 3D trajectory labels are used. Table 8 summarizes the architecture and optimization settings.

Table 8: Architecture and optimization configuration. The reference-sampling ranges apply to SynCamVideo, Ego-Exo4D, and the room- and table-scale subsets of ACE-Data-0.

Setting	Value
Spatial initialization	LagerNVS
Trainable parameters	Motion branch and renderer LoRA
Optimizer	AdamW
Training iterations	16,000
Image resolution	512×512
Reference observations per example	8
Motion bases K	24
Renderer LoRA rank	8
Motion branch learning rate	5×10^{-5}; 500-step warmup, then cosine decay to 10^{-5}
Renderer LoRA learning rate	10^{-6}
Batch size	2 per step; global batch 16 (2×8)
$\lambda_{\mathrm{MSE}}, \lambda_{\mathrm{LPIPS}}, \lambda_{\mathrm{ortho}}$	$1.0,\ 0.5,\ 0.01$
Selected encoder layers	12, 16, and 24
Motion encoder/decoder depths	3 / 2 layers
Attention heads	12
Reference sampling	Random; 16–48-frame temporal span and 2–8-frame gap between adjacent timestamps
Target timestamps	8: four coincide with reference times, four do not

D.2 Evaluation Protocol

Each reference observation is (I_s, t_s, Π_s), and each held-out target is (I^*, t^*, Π^*), where I^* denotes the ground-truth target RGB image. The target RGB image is excluded from the reference set. The same model is used without task-specific finetuning. The main quantitative evaluations cover novel-view synthesis at observed and unobserved timestamps on Neural 3D Video (Li et al., 2022), Immersive (Broxton et al., 2020), and PKU-DyMVHumans (Zheng et al., 2024); synchronized and asynchronous multi-view dynamic NVS on Neural 3D Video; and static multi-view NVS on RealEstate10K (Zhou et al., 2018), DL3DV-10K (Ling et al., 2024), and ACID (Liu et al., 2021). Static observations share one scene time. Synchronized multi-view observations share timestamps across cameras, whereas asynchronous observations may be captured at distinct times. Within a benchmark, comparisons use the same target queries, output resolution, and PSNR, SSIM, and LPIPS implementations. The sampling and aggregation protocols for Tables 1 and 2 are specified below.

Monocular-video evaluation. For Table 1, each example uses eight reference observations, with one camera sampled at each of eight timestamps. The **observed-time** targets comprise the full grid of eight held-out cameras at these same timestamps, yielding 64 queries. The **unobserved-time** targets use the same eight held-out cameras at seven timestamps not present in the reference set but

lying within the reference interval, yielding 56 queries. Both groups use the same reference images and sampling windows. Within each group, metrics are averaged over queries within each example, then over sampling windows within each scene, and finally equally across scenes. For each metric M, Table 1 reports

$$M = \tfrac{1}{2}\left(M_{\text{seen}} + M_{\text{unseen}}\right), \tag{22}$$

giving the two temporal groups equal weight. Table 9 reports the groups separately.

Two Immersive windows with degenerate context-camera alignment are excluded from the common evaluation set for all methods; failed evaluations are not assigned zero scores. To keep the comparison focused on feed-forward reconstruction, C4G is evaluated without VDM, and NeoVerse uses its reconstruction-only output without refinement. These are the configurations used throughout the monocular-video comparison.

Table 9: Separate results at observed and unobserved target times using the same reference images, held-out cameras, and sampling windows. Each window contains 64 observed-time queries and 56 unobserved-time queries.

	Neural 3D Video			Immersive			PKU-DyMVHumans		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Observed target times (seen-time)									
4DGT	12.24	0.414	0.550	11.20	0.347	0.624	17.30	0.714	0.373
C4G (w/o VDM)	14.55	0.465	0.554	14.09	0.505	0.772	16.71	0.529	0.406
MoVieS	18.10	0.652	0.211	13.46	0.435	0.439	20.57	0.571	0.209
NeoVerse (w/o refinement)	11.43	0.322	0.547	10.02	0.185	0.799	16.38	0.627	0.452
NoPo4D	13.19	0.508	0.339	7.50	0.399	0.659	10.41	0.447	0.528
AnySpaceTime	**22.03**	**0.707**	**0.154**	**20.43**	**0.623**	**0.311**	**22.92**	**0.842**	**0.139**
Unobserved target times (unseen-time)									
4DGT	12.35	0.416	0.549	11.22	0.349	0.622	17.33	0.715	0.373
C4G (w/o VDM)	14.55	0.465	0.553	14.08	0.505	0.772	16.72	0.533	0.406
MoVieS	18.14	0.653	0.210	13.46	0.436	0.438	20.52	0.572	0.211
NeoVerse (w/o refinement)	13.37	0.398	0.498	10.95	0.282	0.744	15.95	0.604	0.470
NoPo4D	13.20	0.509	0.339	7.52	0.400	0.657	10.40	0.449	0.529
AnySpaceTime	**22.04**	**0.707**	**0.155**	**20.42**	**0.622**	**0.311**	**22.93**	**0.842**	**0.140**

Synchronized and asynchronous multi-view evaluation. Table 2 evaluates the input patterns in Fig. 3(c,d) on six Neural 3D Video scenes, with five windows per scene and 30 examples per setting. Each reference set contains four cameras $c_0, \ldots, c_3$ and eight images per camera, totaling 32 reference images. For a window starting at frame index t_0, camera c_i samples

$$t_{i,k}^{\text{sync}} = t_0 + 2k$$

in the synchronized setting and

$$t_{i,k}^{\text{async}} = t_0 + i + 2k$$

in the asynchronous setting, where $i = 0, \ldots, 3$ and $k = 0, \ldots, 7$. Synchronized inputs share eight timestamps spanning t_0 to $t_0 + 14$. Asynchronous inputs occupy 18 distinct timestamps from t_0 to $t_0 + 17$, with at most two cameras observed simultaneously.

Both settings use the same camera groups, window start indices, and target queries. Targets form a grid of **eight held-out cameras at eight frame indices**, yielding **64 target images per window**. All target timestamps occur in the reference set, including in the asynchronous setting through camera c_0. Target-camera positions lie within the planar quadrilateral formed by the reference cameras and satisfy a calibration-based coverage filter that approximates image overlap geometrically. Each compared method uses its native inference procedure, provided camera calibration, 512×512 images, and the same checkpoint across both settings. For each setting separately, metrics are averaged over the 64 targets in each window, then over windows within each scene, and finally equally across scenes.

The current asynchronous protocol therefore tests robustness to *temporally misaligned multi-view references*, but does not by itself establish performance when the target timestamp is globally absent from all reference cameras. We state this scope explicitly to avoid over-claiming the "arbitrary observations" setting.

Scope of bullet-time evaluation. We use "bullet-time" to denote the query setting in which the target time is fixed while the target viewpoint varies. Figure 6 of the main paper provides a qualitative evaluation of this setting. We do not introduce a separate bullet-time benchmark or dedicated quantitative metric.

D.3 Reference-Set Size and Temporal Stride

We additionally evaluate monocular dynamic-camera interpolation on DynPose-100K (Rockwell et al., 2025), Neural3DV (Li et al., 2022), and Immersive (Broxton et al., 2020). Here C and T denote the numbers of reference and target frames, respectively, and stride denotes the frame-index spacing used to sample the sequence. Table 10 varies C on DynPose-100K while holding $T = 3$ and stride at two. Because increasing C under this protocol also widens the temporal interval covered by the references, this experiment measures the *combined* effect of reference count and temporal coverage. It should not be interpreted as a controlled reference-count ablation. Table 11 separately varies stride on Neural3DV and Immersive while holding $C = T = 8$.

Table 10: Coupled effect of reference-set size and temporal coverage on DynPose-100K for monocular dynamic-camera interpolation. All rows use the same three target frames and stride two. Increasing C also widens the temporal interval covered by the references; therefore this table should not be interpreted as isolating the causal effect of reference count alone.

Reference frames C	Target frames T	PSNR ↑	SSIM ↑	LPIPS ↓
4	3	23.129	0.7606	0.1594
16	3	22.649	0.7369	0.1727
32	3	21.601	0.7227	0.2116

The target frames are identical across all three settings. PSNR decreases from 23.129 to 21.601 dB as the reference set grows from 4 to 32 frames, a difference of 1.528 dB. Because the added references also extend the observed temporal interval, the trend reflects the joint effect of a larger reference set and a wider temporal span rather than the isolated effect of either factor.

Table 11: Temporal stride for monocular dynamic-camera interpolation with eight reference and eight target frames. Each entry is PSNR.

Dataset	Samples	Stride 2	Stride 8	Stride 16
Neural3DV	30	29.626	27.579	26.560
Immersive	28	26.964	26.128	25.453

PSNR decreases as the sampling stride increases on both datasets, indicating that wider temporal gaps are harder for the model under the reported protocol.

D.4 Inference Time and Peak Memory

Efficiency depends on both the number of reference observations S and the number of target queries Q. Table 12 separates reference encoding from target adaptation and rendering, and reports end-to-end latency and peak allocated device memory for each (S, Q) setting.

Reference encoding time and peak memory increase with S. At fixed S, encoding time is nearly unchanged between one and eight targets, while target-stage time grows by approximately a factor of eight. Peak memory is identical for the two target counts at each measured reference-set size. At $S = 8$, the amortized end-to-end latency decreases from 1.338 s per target at $Q = 1$ to 0.343 s per target at $Q = 8$.

Table 12: Measured inference time and peak memory as the numbers of reference observations S and target queries Q vary. Encoding and target-stage latencies are shown separately; end-to-end latency and peak allocated memory cover the full inference call.

S	Q	Encode (s)	Targets (s)	End-to-end (s)	Peak memory (GB)
4	1	0.585	0.128	0.714	6.71
4	8	0.584	1.029	1.612	6.71
8	1	1.137	0.202	1.338	8.01
8	8	1.137	1.611	2.747	8.01
16	1	2.460	0.362	2.822	10.59
16	8	2.460	2.888	5.347	10.59
32	1	6.535	0.688	7.223	15.75
32	8	6.535	5.516	12.052	15.75

These measurements characterize the scaling behavior of AnySpaceTime itself. Without matched measurements of competing methods under the same hardware, resolution, reference count, and target count, we do not use this table to claim a runtime advantage over the baselines. The purpose of Table 12 is to characterize amortization and scaling with S and Q, rather than to rank methods by inference efficiency.

D.5 Additional Notes on Static-Scene Evaluation

For the static multi-view experiments, the purpose is to test whether temporal modeling substantially degrades the spatial capability inherited from the pretrained LagerNVS backbone. Table 3 of the main paper shows that AnySpaceTime remains close to LagerNVS across all three static-scene benchmarks, but does not uniformly improve every metric. On RealEstate10K, PSNR changes from 29.85 to 29.51 and SSIM from 0.899 to 0.894, while LPIPS improves from 0.075 to 0.068. On DL3DV, PSNR changes from 19.65 to 19.18 and SSIM from 0.601 to 0.581, while LPIPS improves from 0.321 to 0.313. On ACID, PSNR changes from 30.89 to 30.59 and SSIM from 0.860 to 0.856, while LPIPS improves from 0.122 to 0.102.

We therefore interpret the static-scene results as evidence that introducing temporal adaptation largely preserves the pretrained static NVS capability, rather than as evidence that the temporal branch uniformly improves static-scene reconstruction. This interpretation is also consistent with the evaluation setting in which the temporal branch remains active at test time.

The unusually low LVSM PSNR reported on DL3DV cannot be explained from the information available in the current manuscript alone. We therefore do not attribute it to a specific cause here.