BaLi-RF: Bandlimited Radiance Fields for Dynamic Scene Modeling

Sameera Ramasinghe Violetta Shevchenko Gil Avraham Anton Van Den Hengel
Amazon, Australia

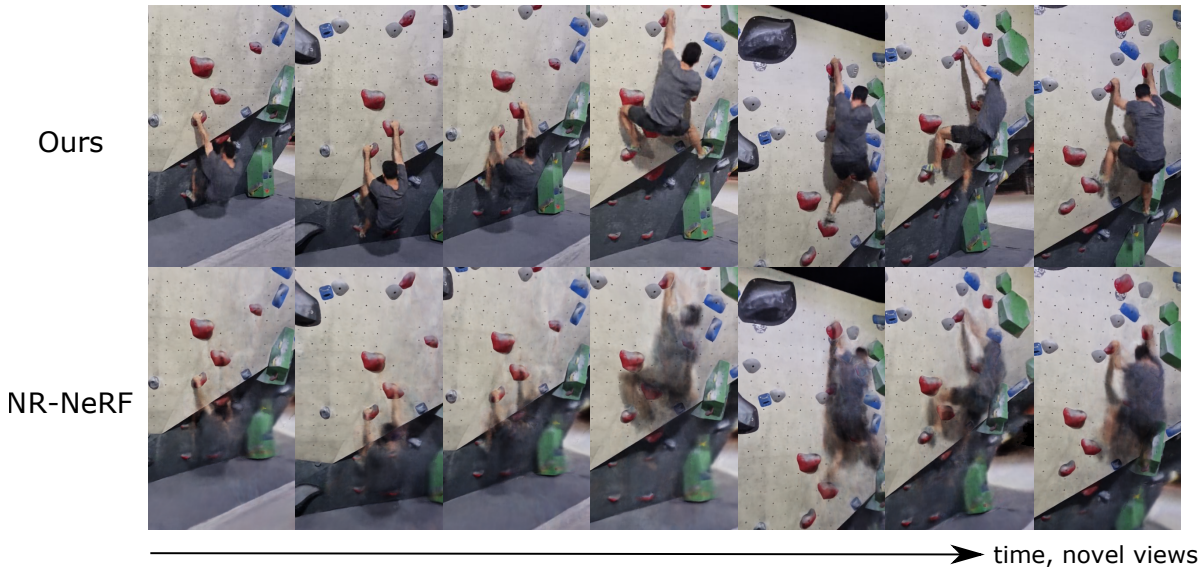

Figure 1. We address the problem of recovering the 3D structure of a dynamic scene given sparse RGB views from a monocular, moving camera. The figure shows a comparison between the synthesized novel views of a challenging scene with long-range dynamics, against a competitive baseline. As illustrated, our model is able to better capture the fine details and accurately localize the motion compared to NR-NeRF [45]. We attribute the superior performance of our model to efficient factorization of time and space dynamics that enable incorporating well-defined spatio-temporal priors, leading to better recovery of complex dynamics.

Abstract

Reasoning the 3D structure of a non-rigid dynamic scene from a single moving camera is an under-constrained problem. Inspired by the remarkable progress of neural radiance fields (NeRFs) in photo-realistic novel view synthesis of static scenes, extensions have been proposed for dynamic settings. These methods heavily rely on neural priors in order to regularize the problem. In this work, we take a step back and reinvestigate how current implementations may entail deleterious effects, including limited expressiveness, entanglement of light and density fields, and sub-optimal motion localization. As a remedy, we advocate for a bridge between classic non-rigid-structure-from-motion (NRSfM) and NeRF, enabling the well-studied priors of the former to constrain the latter. To this end, we propose a framework that factorizes time and space by formulating a scene as a composition of bandlimited, high-dimensional signals. We demonstrate compelling results across complex dynamic scenes that involve changes in lighting, texture and long-range dynamics. Our codes and data will be released.

1. Introduction

The problem of scene modeling [13, 23] is a prominent pillar in the field of computer vision with applications ranging from novel view synthesis [3, 10], augmented and virtual reality [5, 7], SLAM [19, 30], and many more. Particularly under static scene conditions, NeRF [29] has recently exhibited remarkable progress in synthesizing photorealistic novel views from sparse 2D images.

The hallmark of NeRF is the architectural bias of neural networks. That is, the natural (Lipschitz) smoothness of neural functions acts as an implicit *neural prior*. This property imposes self-regularization [8, 22, 41] to otherwise ill-posed problems [55]. Recently, multiple works have at-

tempted to extend NeRF to dynamic settings [14, 15, 21, 28, 36, 45, 48, 50, 51], leveraging these neural priors that made NeRFs successful. However, real-world dynamic scenes generally violate the analytical conveniences of multi-view geometry [20]. In this vein, dynamic NeRF works have primarily resorted to using ray deformation paradigms [31] for addressing geometric inconsistencies [14, 27, 35, 36, 45, 50]. Although these approaches have yielded remarkable results, we show that their over-reliance on the smoothness of neural priors cause fundamental problems; using a neural network to simultaneously model both time and space is detrimental to accurate scene modeling, as space typically consists of sharp/high-frequency details, whereas temporal dynamics are naturally smooth and continuous (see Sec. 3).

On the other hand, the roots of dynamic scene modeling more classically extend to the problem of non-rigid structure from motion (NRS*f*M). In summary, NRS*f*M concerns recovering sparse 3D point deformations of a scene from 2D point correspondences between multiple 2D projections. Similar to the dynamic NeRF, NRS*f*M setting is also severely underconstrained. In contrast to NeRF, however, NRS*f*M literature is heavily focused on formulating explicit priors to convert this ill-posed problem to a well-defined one. The performance of NRS*f*M models mainly depends on the alignment of these priors with the deformation in question. Thus, since the early work of Bregler *et al*. [6], which presented a classic row-rank factorization approach, a plethora of studies have explored different priors on shape space [37, 43, 44], point trajectories [2, 17, 18], or subspaces [1, 25].

The central thesis of this paper aims at presenting a generic framework to combine the strengths of implicit neural priors of NeRFs and well-designed explicit priors that are deeply rooted in NRS*f*M literature. To this end, we model the light and density fields of a 3D scene as bandlimited, high-dimensional signals. This particular standpoint enables complete factorization of spatio-temporal dynamics, allowing us to inject explicit priors on the time and space dynamics independently. To demonstrate the practical utility of our framework, we offer an example implementation that enforces 1) a low-rank constraint on the shape space, along with 2) a neural prior and 3) a union-of-subspace prior on the time space. We show that the strong regularization effects entwined with these priors enable our model to reconstruct long-range dynamics and localize motion accurately, only using sparse RGB images for supervision. Our contributions are three-fold:

- We show that existing mainstream extensions of NeRF to dynamic scenes suffer from critical drawbacks, primarily due to their over-reliance on neural priors.
- We propose a generic framework that enables full factorization of space and time by formulating radiance fields as bandlimited signals. We only utilize RGB images from a monocular camera for supervision.
- We empirically validate the efficacy of our framework by demonstrating better modeling of long-range dynamics, motion localization, and light/texture changes, compared to the baselines with more than $10\times$ faster training times.

2. Related Work

Most successful methods for modeling dynamic scenes require either a setup containing multiple cameras [11, 12, 46, 56, 57] or active depth sensors [31, 32, 42, 53]. In contrast, recovering the 3D structure of a scene using a monocular camera is a more challenging task that has been approached from various angles [4, 12, 27, 32, 33, 49, 52]. However, this paper only focuses on NeRF extensions and NRS*f*M.

NRS*f*M. The problem of NRS*f*M focuses on modeling the 3D structure of sparse points using their 2D projections. To convert this problem to a well-defined one, various priors have been explored. These priors can be mainly categorized as shape-based and trajectory-based priors.

Breger *et al*. [6], in their seminal work, argued that NRS*f*M could be solved using a finite number of low-rank shape-basis functions [16]. Later, Torrensani *et al*. [44] modeled the coefficients of the shape-basis as a linear dynamical system. In contrast, Rabaud *et al*. [37] proposed to learn a smooth manifold of shape configurations, and Gotardo *et al*. [17] explored non-linear shape models using kernels. More recently, Agudo *et al*. [1] imposed a union-of-subspace prior to constrain the shape deformations. Another interesting work revealed that learning shape deformations can be formulated as a block sparse dictionary learning problem [24]. Considering trajectory-based priors, Akhter *et al*. [2] demonstrated that instead of decomposing the shape deformation over time with basis functions, the trajectory of measurements could be formulated as DCT basis functions. In the same spirit, [60] exploited the convolutional structure of the trajectories. Multiple works [1, 25, 54, 59] showed that frames could be clustered to restrict trajectories within low-dimensional subspaces. This closely aligns with the manifold prior we propose in Sec. 4.5. Multiple works have also sought to explicitly regularize trajectories by minimizing their response to high-pass filters [47], injecting rigid key-frames [58], enforcing sparsity priors [40], and considering articulated motion [34]. Nonetheless, NRS*f*M typically deals with sparse 3D points; in contrast, we focus on novel view synthesis, which requires reasoning dense 3D structure.

Dynamic NeRF. Inspired by the success of NeRF, many studies have attempted to model dynamic neural radiance fields using the concept of ray deformation [14, 27, 35, 36, 45]. D-NeRF [36] was the first among the above to propose a general framework which learns a displacement per continuous point, from a given radiance field to a canonical

one. Both [14, 45] extended this idea, and further introduced a constraint to model the foreground and background separately, thus allowing quicker convergence and a better-constrained search space. [14] introduced a method to disambiguate self-occlusions that hinders the performance of these approaches. Nerfies [35] achieves remarkable results on novel views synthesis of dynamic scenes by incorporating elastic regularization, but specifically target self-portraits. Finally, several other NeRF extensions have also been proposed that require depth estimates [50], optical flows [48], foreground masks [14, 21], meshes [51], or assume that dynamic objects are distractors to remove [28]. In Sec. 3, we critically analyze ray deformation approaches, and in fact, show that these methods model deformations of the light and density fields over time, instead of rays.

3. Revisiting ray deformation networks

Extending NeRF to dynamics scenes fundamentally involves representing the scene as a continuous function with 6D inputs $(x, y, z, \theta, \phi, t)$, where t is the time and (θ, ϕ) is the viewing direction. However, it has been empirically validated [36] that employing a single MLP that learns a mapping from 6D inputs to density and color fields yields sub-optimal results. Hence, existing works decompose the aforementioned task into two modules [14, 27, 35, 36, 45]: 1) the first MLP learns a warping field of 3D points $(\Delta x, \Delta y, \Delta z)$ sampled along the rays with respect to a canonical setting; 2) the second module then acts similarly to the original NeRF formulation, regressing the density and light fields given the warped samples along the rays $(x + \Delta x, y + \Delta y, z + \Delta z)$. Since the warping is applied to points sampled along the ray, this formulation is interpreted as deforming the rays as a function of time. Further, note that a rudimentary assumption here is that the objects do not enter or leave the scene, and the lighting/texture is consistent. However, we notice that existing implementations of this framework do not adhere to these constraints (see Supp. A). Specifically, we show that such networks can indeed model light and density changes separately (to an extent), which is infeasible with a model that only learns ray deformations (see Fig. 4 and Fig. 3). However, to avoid confusion, we will keep referring to this class of models as ray deformation networks. Next, we discuss several critical limitations of them.

3.1. Limitations of ray deformation networks

In this section, we present a brief exposition of the limitations entailed with ray deformation networks. For an extended analysis, refer to Supp. A.

Dependency on a canonical frame: Ray deformation networks require choosing a canonical frame arbitrarily, where most models commonly choose the frame at $t = 0$ to this end. However, this choice can significantly harm the model performance in cases where 1) objects or the camera are subjected to long-range translations, and 2) new objects appear in future. In both cases, the canonical frame at $t = 0$ needs to learn an average scene representation where all future information is present, which becomes increasingly infeasible as the scene becomes more complex. On the other hand, the model also needs to preserve the continuity; the model output at $(t = \delta t)$ needs to be a smooth transition of the canonical scene at $t = 0$, which can be impractical if the scene comprises abundant future information. In contrast, our framework does not depend on a canonical scene.

Entanglement of light and density fields: Although ray deformation networks are able to deform the light and density fields, they are still highly entangled. More precisely, it can be shown that in order to achieve complete disentanglement of the light and density fields, the network needs to preserve a specific block-diagonal Jacobian structure in one of the hidden layers, which is an extremely restrictive requirement. In comparison, our framework achieves complete disentanglement by design, modeling the light and density fields independently.

Limited expressiveness: Ray deformation networks comprise a bottleneck of dimension three. Therefore, each of the density and light fields modeled by this network becomes three dimensional manifolds. Thus, they cannot encode complex dynamics that needs to be parameterized by four variables (x, y, z, t) simultaneously.

Substandard separation of background and motion: Ray deformation networks model the warp field using a single MLP. However, this is a substandard design choice since the space and time variations have different spectral properties. For instance, the space may contain high-fidelity details, and in contrast, time dynamics are generally smoother. Therefore, using an MLP with a particular bandwidth for learning both spatial and time variations together leads to sub-optimal reconstructions. On the contrary, our framework enables factorization of space and time dynamics, allowing better separation of static and dynamic regions. Next, we formally present our framework.

4. Our framework

Consider a set of 2D projections $\{I(t_n)\}_{n=1}^{N}$ of a 3D scene captured from a moving camera. For brevity, we drop the dependency on the camera poses from the notation. Without the loss of generality, we assume that the scene is bounded within a cube with side length D. We begin by observing that there exists a latent density and color field corresponding to each $I(t_n)$, which can be discretized into a cubic grid of D^3 nodes. Then, rewriting the latent states of either field in the matrix form, gives

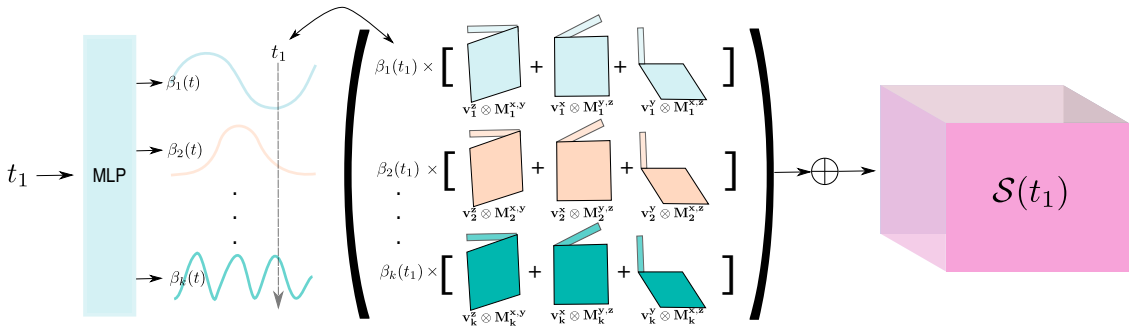

Figure 2. **The proposed implementation of our framework.** We treat the light and density fields as bandlimited, high-dimensional signals (only a single field is shown in the figure). The time evolution of each 3D point (x, y, z) of the field is modeled as a finite linear combination of time-basis functions $\{\beta_j(t)\}$. The coefficients of the $\{\beta_j(t)\}$ are decomposed into outer products between learnable matrices ($\mathbf{M}$) and vectors ($\mathbf{v}$). This decomposition is inspired [9]. Our formulation allows efficient factorization of time and space dynamics, leading to high-quality reconstructions of complex dynamics, along with faster convergence.

$$\mathbf{S} = \begin{bmatrix} s(t_1, x_1, y_1, z_1) & s(t_1, x_2, y_2, z_2) & \dots & s(t_1, x_{D^3}, y_{D^3}, z_{D^3}) \\ \vdots & \vdots & \ddots & \vdots \\ s(t_N, x_1, y_1, z_1) & s(t_N, x_2, y_2, z_2) & \dots & s(t_N, x_{D^3}, y_{D^3}, z_{D^3}) \end{bmatrix}_{N \times D^3} \tag{1}$$

where $s(t_i, x_i, y_i, z_i)$ can be either the density or the emitted color values at (x_i, y_i, z_i) point at time t_i.

4.1. Memorization of latent states

Let $\text{rank}(\mathbf{S}) = K \leq N$ (assuming $N < D^3$). Then, there exist K basis vectors, each with dimension D^3, that can perfectly reconstruct (memorize) $\mathbf{S}$. More precisely, in this case, each row of $\mathbf{S}$ can be reconstructed as

$$\mathbf{S}(t_i, \cdot) = \sum_{j=1}^{K} a_j(t_i)\hat{\boldsymbol{\alpha}}_j, \tag{2}$$

where $\mathbf{S}(t_i, \cdot)$ is the i^{th} row of $\mathbf{S}$, $\{\hat{\boldsymbol{\alpha}}_j\}_{j=1}^{K}$ are basis vectors of dimension D^3, and $\{a_j\}_{j=1}^{K}$ are scalar coefficients. Intuitively, each row of $\mathbf{S}$ corresponds to a snapshot of the field in space at a particular time instance. On the contrary, each column of $\mathbf{S}$ are snapshots of the time evolution of a particular (x, y, z) point in the field. We note an interesting duality here; since the dimension of the row space and the column space of $\mathbf{S}$ are equal, it should be possible to reconstruct the time evolution of the density/color value of each (x, y, z) position also using a K number of basis vectors. Thus, we model the time evolution of each point as

$$\mathbf{S}(x_i, y_i, z_i, \cdot) = \sum_{j=1}^{K} b_j(x_i, y_i, z_i)\hat{\boldsymbol{\beta}}_j, \tag{3}$$

where $\{\hat{\boldsymbol{\beta}}_j\}_{j=1}^{K} \in \mathbb{R}^N$ are basis vectors and $\{b_j\}_{j=1}^{K}$ are scalars. This change of perception is crucial for generalizing to unseen time instances and obtaining a space-time factorization, as we show in Sec. 4.2.

4.2. Bandlimited fields and generalization

In Sec. 4.1, we established that the imposition of a low-rank assumption on the time-evolving field allows us to recover a set of observations using a finite number of basis vectors. However, recall that, in practice, only a sparse set of 2D observations $\{I(t_n)\}_{n=1}^{N}$ are at our disposal. Therefore, memorization is not sufficient, and the framework should be able to generalize to unseen time instances. To achieve this, we employ an important assumption here that the *light and density fields are bandlimited signals*. [1] This particular assumption enables us to convert $\{\hat{\boldsymbol{\beta}}_j\}$ in Eq. 3 to continuous time-dependent functions, thereby obtaining a continuous time-evolving field,

$$\mathbf{S}(x, y, z, t) = \sum_{j=1}^{K} \beta_j(t) b_j(x, y, z). \tag{4}$$

Observe that under this view, $\{\hat{\boldsymbol{\beta}}_j\}$ can be considered as discrete samples of the continuous functions $\{\beta_j(t)\}$. Further, Eq. 4 provides a nice factorization of time and spatial dynamics, allowing us to impose priors on time and space independently. In Sec. 4.3, we present an implementation of the proposed framework. In this implementation, we inject a low-rank prior on space, along with smoothness and compact manifold priors on time. It is worth to note that our framework is generic enough to support alternative implementations and more complex priors, which we leave to future explorations.

4.3. Implementation

Leveraging the factorization we achieved in Eq. 4, we can formulate the entire 3D field volume as a time-dependent higher dimensional signal, that can be decomposed into a linear combination of 3D tensors $\mathcal{A}_j^{xyz} \in$

[1] Note that we use the general notion of bandlimitedness here; a signal is bandlimited if, and only if, it can be reconstructed using a finite set of basis functions.

$\mathbb{R}^{D\times D\times D}$:

$$\mathcal{S}(t) = \sum_{j=1}^{K} \beta_j(t)\mathcal{A}_j^{xyz}, \tag{5}$$

where $\mathcal{S}(t) \in \mathbb{R}^{D\times D\times D}$ is the state of the field at time t. Note that we adopt the tensor notation here where the superscripts denote the dimensions, *i.e.*, $x = 1,\dots D, y = 1,\dots,D$, and $z = 1,\dots,D$. To regularize the spatial variations, we employ a low-rank constraint on $\mathcal{A}_j$ as,

$$\mathcal{S}(t) = \sum_{j=1}^{K} \beta_j(t)(\mathbf{v}_j^z \otimes \mathbf{M}_j^{xy} + \mathbf{v}_j^x \otimes \mathbf{M}_j^{yz} + \mathbf{v}_j^y \otimes \mathbf{M}_j^{xz}), \tag{6}$$

where $\mathbf{v}_j \in \mathbb{R}^D$ and $\mathbf{M}_j \in \mathbb{R}^{D\times D}$ are one- and two-dimensional tensors, respectively, and $\otimes$ is the outer product. The above choice of factorization is inspired by the *VM-decomposition* proposed in [9]. This factorization accomplishes two goals: 1) enforcing a low rank constraint on the spatial variations of the field, and 2) significantly reducing the size of the model and the number of trainable parameters. We note that such low-rank priors have been widely employed in the NRS*f*M literature for the same purpose [37, 43, 44].

4.4. Neural trajectory basis

In theory, it is possible to use any class of functions that form a complete basis in $L^2(\mathbb{R}, dt)$ as $\{\beta_j(t)\}$. Several such popular choices include the DCT basis, Fourier basis, and Bernstein basis, among many others. Nonetheless, we use neural networks to parameterize our basis functions, leveraging the implicit architectural smoothness constraint built into them. We dub these basis functions as *neural trajectory basis*. Neural trajectory basis present an important, implicit prior to our model that the field values should evolve smoothly. We also empirically noted that neural basis functions are naturally more expressive and adaptive as they are learned end-to-end, as opposed to other choices (see Table 2). Expressiveness is crucial, as it is desirable to model the dynamics of each point with a minimal number of basis functions. Thus, we compute $\{\beta_j(t)\}$ via an MLP $\mathcal{F}(t) : \mathbb{R} \to \mathbb{R}^K$ as,

$$\mathcal{F}(t) = [\beta_1(t), \beta_2(t), \dots, \beta_K(t)]. \tag{7}$$

We also show that the smoothness prior embedded into the neural trajectory basis closely aligns with the work of Valmadre *et al*. [47], where they showed that, in NRS*f*M, trajectory's response to high-pass filters should be minimal. We validate that neural trajectory basis implicitly preserve this property (see Supp. C).

4.5. Manifold Regularization

Multiple works in NRS*f*M have explored restricting the subspace of dynamics in order to obtain better reconstructions. The high-level objective is to temporally cluster the motion in order to restrict similar dynamics into a low-dimensional subspace [1, 25, 54, 59]. We observed that such a constraint can improve our reconstructions also. More formally, we empirically asserted that better results are obtained by locally restricting the dimension of the submanifold that $\mathcal{S}(t)$ is immersed in. Instead of clustering the motion across the entire sequence, we assume that dynamics are locally compact: movements that occur over a small time period can be described using a smaller subspace. To enforce this constraint, we adopt the following procedure.

Observe that $\frac{\partial \mathcal{S}(t)}{\partial t}$ exists for all t. Also, for a scene with (at least locally) continuously deforming light and density fields, we make the fair assumption that there exists a bijection from the time domain to $\mathcal{S}(t)$, *i.e.*, $\mathcal{S}(t_1) = \mathcal{S}(t_2) \Leftrightarrow t_1 = t_2$. Further, the space $\mathcal{S}(t)$ is a Hausdorff space and the domain of $\mathcal{S}(t)$ is compact. Recall the following theorem.

Theorem: *Continuous bijection from a compact space to a Hausdorff space is a homeomorphism.*

Therefore, $\mathcal{S}(t)$ is a 1-dimensional manifold embedded in a D^3-dimensional space, and its local coordinate chart is a compact subspace in $\mathbb{R}$. Further, at any given time t, $\mathcal{S}(t)$ is a linear combination of K points $\{\mathbf{v}_j^z \otimes \mathbf{M}_j^{xy} + \mathbf{v}_j^x \otimes \mathbf{M}_j^{yz} + \mathbf{v}_j^y \otimes \mathbf{M}_j^{xz}\}_{j=1}^K \in \mathbb{R}^{D\times D\times D}$. Therefore, $\mathcal{S}(t)$ is a submanifold of $\mathbb{R}^K$.

Now, let $\mathbf{P}_j^{xyz} = (\mathbf{v}_j^z \otimes \mathbf{M}_j^{xy} + \mathbf{v}_j^x \otimes \mathbf{M}_j^{yz} + \mathbf{v}_j^y \otimes \mathbf{M}_j^{xz})$. Suppose the dimension of the local submanifold we need is W, such that $K = dW$ for some integer d. Then, we define the 4D tensor $\mathbf{Q}_{j:j+W}^{xyzu} \in \mathbb{R}^{D\times D\times D\times W}$ such that $\mathbf{Q}_{j:j+W}^{xyzu} = \{\mathbf{P}_u^{xyz}\}_{u=j}^{j+W}$. Next, we obtain

$$\tilde{\mathbf{Q}}^{xyzu}(t) = \sum_{n=0}^{d-1} \mathbf{Q}_{(nW+1):W(n+1)}^{xyzu} \odot \mathrm{sinc}\big((d-1)(t - \tfrac{n}{(d-1)})\big), \tag{8}$$

where $\odot$ is the element-wise multiplication, and $\mathrm{sinc}(r) = \begin{cases} 1, & \text{if } r = 0 \\ \frac{\sin(r)}{r}, & \text{otherwise} \end{cases}$. The choice of the sinc function here is not arbitrary, and is crucial for the smooth transition between submanifolds as the time progresses. More precisely, the sinc interpolation ensures that no higher frequencies than $(d-1)/2$ can be presented in $\tilde{\mathbf{Q}}^{xyzu}(t)$ along the temporal dimension. Finally, we can obtain the regularized field as,

$$\tilde{\mathcal{S}}(t) = \sum_{u=1}^{W} \beta_u(t)\tilde{\mathbf{Q}}^{xyzu}(t). \tag{9}$$

From a strict theoretical perspective, one can argue that Eq. 9 violates the time and space factorization we obtained in Eq. 6. However, in practice, the sinc interpolation ensures that $\tilde{\mathbf{Q}}^{xyzu}(t)$ is locally almost constant as long as we choose d to be suitably small, as $\tilde{\mathbf{Q}}^{xyzu}(t)$ cannot then have higher frequencies than $(d-1)/2$. Further, Eq. 9 ensures that $\tilde{\mathcal{S}}(t)$ can only locally traverse within an $\mathbb{R}^W$ subspace where $W < K$, which is a more regularized setting than

Eq. 6, where $\mathcal{S}(t)$ is allowed to traverse within an $\mathbb{R}^K$ subspace.

4.6. Volume Rendering

Let us denote $\mathcal{C}_{\mathbf{x}}(t)$ and $\mathcal{Z}_{\mathbf{x}}(t)$ as continuously evolving light and density fields, respectively, obtained via Eq. 9 and queried at 3D position $\mathbf{x}$. We can obtain density and light values at any $\mathbf{x}$ at time t as,

$$\sigma(\mathbf{x}, t), c(\mathbf{x}, t) = \mathcal{Z}_{\mathbf{x}}(t), \mathcal{C}_{\mathbf{x}}(t). \tag{10}$$

To compute the above values at an arbitrary continuous position $\mathbf{x}$, we tri-linearly interpolate the grids. Then, the rendering is done similarly to the original NeRF formulation: let $\mathbf{x}(h) = \mathbf{o} + h\mathbf{d}$ be a 3D location sampled on the ray emitted from camera center $\mathbf{o}$ in the direction of $\mathbf{d}$, passing through a pixel p. We can obtain the predicted pixel color $\tilde{p}$ at a given time instance t as,

$$\tilde{p}(t) = \int T(\sigma(\mathbf{x}(h), t), h)\sigma(\mathbf{x}(h), t)c(\mathbf{x}(h), t)dh \tag{11}$$

where $T(\cdot) = \exp\big(-\int_{-\infty}^{\mathbf{x}(h)} \sigma(\mathbf{x}(h), t)dh\big)$. We use the same discrete approximations used in [29] for the above formulas in practice. The final loss $\mathcal{L}$ used for training is the mean squared loss between p and $\tilde{p}$, along with a total variation (TV) loss spatially applied across grid values:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|p(t) - \tilde{p}(t)\| + \lambda_1 TV(\mathcal{Z}(t)) + \lambda_2 TV(\mathcal{C}(t)). \tag{12}$$

Two important remarks are in order: *a*) our model only requires the TV loss as a loss regularizer, as opposed to multiple explicit regularizations that are used in many existing dynamic NeRF architectures such as explicit foreground-background modeling [14, 45], energy-preservation [35], or temporal consistency losses [27, 48]. *b*) To address the insufficiency of neural priors in regularizing the architecture, many dynamic NeRF methods tend to adopt cumbersome training procedures to converge to a good minimum, *e.g.*, sequential training of temporally-ordered frames [27, 36], coarse-to-fine annealing of hyperparameters [35], or morphology processing [52]. In contrast, we simply randomly sample points in time and space and feed them to the model for training. We argue that this is a strong indicator of the well-built inductive bias/implicit regularization of our architecture and the stability of our formulation.

5. Experiments

In this section, we empirically validate the efficacy of our proposed framework.

Datasets: We collect four synthetic scenes and four real-world scenes as our dataset. The synthetic scenes include texture changes, lighting changes, scale changes, and long-range movements. Similarly, the real-world scenes include lighting changes, long-range movements, and spatially concentrated dynamic objects. All the scenes consist of RGB images captured from a single moving camera along with camera poses. For more details on our datasets, see Supp. D.

Baselines: We choose D-NeRF [36] and NR-NeRF [45] as our main baselines. Both are recently proposed Dynamic-NeRF models that adopt the ray deformation paradigm, and only utilize RGB images from a monocular camera for supervision. NR-NeRF architecture comprises an explicit neural network for isolating the motion of a scene, which provides an ideal baseline to evaluate the efficacy of the space-time priors in our model. For above models, we performed a grid search for the optimal hyperparameters for each scene, for fair comparison. In contrast, our model uses a single hyperparameter setting across all the scenes, demonstrating its robustness. Further, it is essential to precisely validate whether the superior performance of our model stems from the light/density disentanglement or the space-time factorization. Therefore, we design another baseline T-TensoRF, which disentangles the light and density fields, but do not factorize time and space dynamics (see Supp. F).

5.1. Synthetic scenes

The synthetic scenes consist of four scenes: *texture change, falling and scale, light move, and ball move*. See Fig. 3 for a qualitative comparison. As shown, D-NeRF and NR-NeRF fail to accurately model the color and light changes. This is an illustration of our claim in Sec. 3, that for full disentanglement of light and density fields, the above methods require a block diagonal Jacobian structure, which is an extremely restrictive condition. Similarly, D-NeRF and NR-NeRF both tend to deform the objects when scale changes and long-range movements are present. T-TensoRF, due its ability to disentangle light and density fields, adequately recovers light/texture changes. However, it depicts inferior performance in falling and scale, and ball move scenes. Further, note that all the baselines fail to accurately learn the 3D positions of the objects showcasing their inability to precisely disentangle camera and scene dynamics. In comparison, our method achieves significantly superior results in all above aspects. See Table 1 for quantitative results.

5.2. Real-world scenes

The real-world scenes contain four challenging scenes; *cat walking, flashlight, flower, and climbing*. Cat walking and climbing scenes contain long-range movements. See Fig. 4 and Fig. 1 for qualitative comparisons on these scenes. As evident, when long-range movements are present, D-NeRF, NR-NeRF, and T-TensoRF fail to recover the high-fidelity details of the moving objects. Similarly, in the flashlight scene, the above methods fail to accurately capture granular details in the background. In the flower

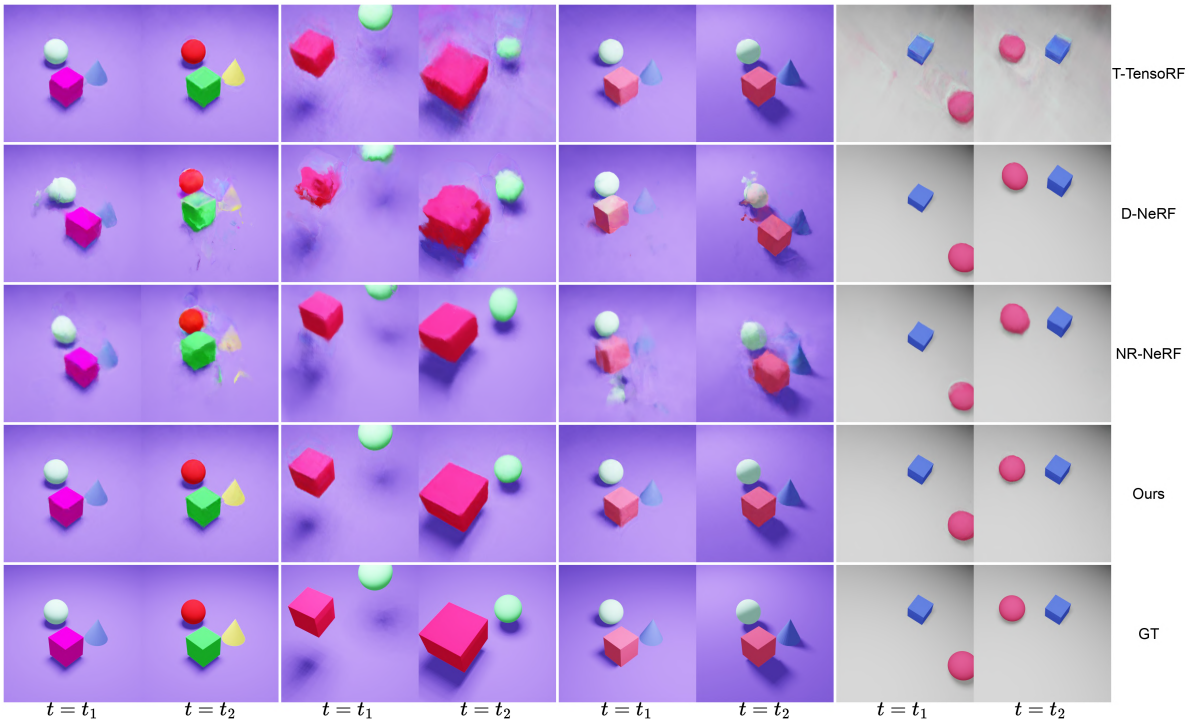

Figure 3. **Qualitative comparison on the synthetic dataset.** The shown reconstructions are from novel views at unseen time instances. As evident, both D-NeRF and NR-NeRF fail to accurately infer the 3D structure of the scenes containing texture and lighting changes (columns $1, 2, 5, 6$). This behavior is caused by their inability to precisely disentangle light and density fields (see Sec. 3.1). In contrast, T-NeRF performs relatively well in these scenes as it achieves this disentanglement by construction. However, all the three baselines exhibit poor reconstructions in the scale change and ball move scenes (columns $3, 4, 7, 8$). This is an illustration of the sub-optimal localization of motion caused by inferior factorization of time and space, that are built into these models. Further, note that the objects in all the scenes are slightly misaligned in baseline reconstructions, demonstrating sub-par disentanglement between scene and camera dynamics. In comparison, our model is yields significantly better reconstructions, demonstrating its better formulation with respect to the above aspects.

Method	Cat			Climbing			Flashlight			Flower		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TensoRF	24.05	0.81	0.35	22.53	0.76	0.34	26.70	0.90	0.36	26.36	0.86	0.29
T-TensoRF	29.45	0.88	0.24	27.08	0.81	0.30	28.93	0.91	0.31	27.10	0.85	0.33
D-NeRF	27.49	0.86	0.30	28.90	0.85	0.27	31.79	**0.95**	0.23	28.56	**0.90**	0.25
NR-NeRF	26.63	0.82	0.35	25.59	0.79	0.35	30.59	0.93	0.29	25.57	0.84	0.37
BaLi-RF (ours)	**29.69**	**0.89**	**0.21**	**29.14**	**0.86**	**0.25**	**31.80**	**0.95**	**0.19**	**29.98**	**0.90**	**0.22**

Method	Color Change			Falling and Scale			Light Move			Ball Move		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TensoRF	19.08	0.89	0.22	17.30	0.86	0.35	19.61	0.79	0.47	24.31	0.94	0.28
T-TensoRF	35.16	**0.97**	0.09	24.30	0.90	0.32	36.49	0.97	**0.10**	28.27	0.96	0.33
D-NeRF	17.42	0.89	0.28	24.60	0.92	0.23	19.15	0.91	0.25	22.58	0.95	0.20
NR-NeRF	16.37	0.89	0.27	15.97	0.86	0.26	18.55	0.91	0.26	23.21	0.95	0.21
BaLi-RF (ours)	**36.68**	**0.97**	**0.08**	**35.74**	**0.97**	**0.11**	**38.04**	**0.98**	**0.10**	**39.32**	**0.99**	**0.09**

Table 1. Quantitative comparison of novel view synthesis on our real and synthetic datasets. We report the average PSNR, SSIM (higher is better) and LPIPS (lower is better) results. Best results are in bold.

scene, where the dynamics are concentrated spatially, all the baselines perform fairly well. On the contrary, our method depicts better results with respect to all aforementioned aspects. Interestingly, note that D-NeRF and NR-NeRF both can model lighting changes as shown in the flashlight scene (see also Supp. G). This validates our insights in Sec. 3 that so-called ray deformation models indeed encode density and light field dynamics, instead of learning ray deformations. See Table 1 for quantitative results.

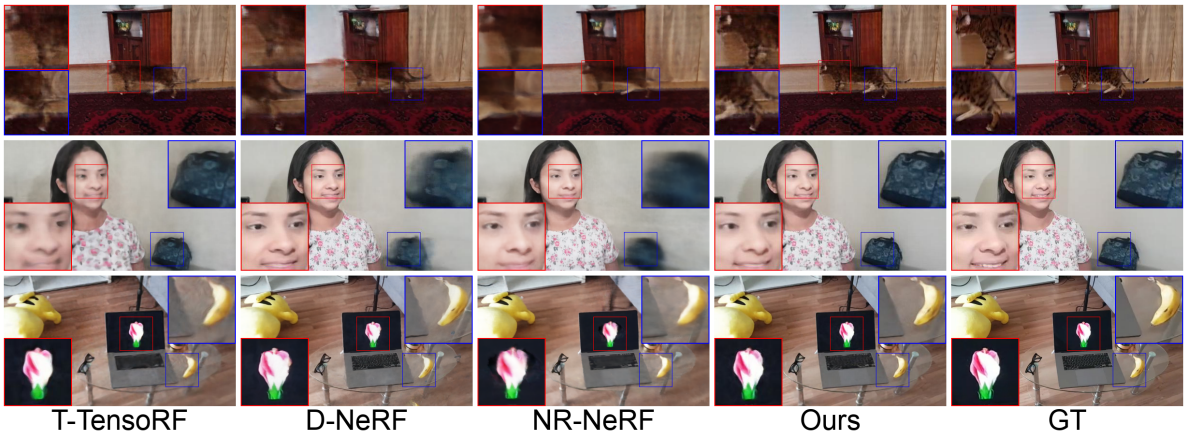

Figure 4. **Qualitative comparison on the real-world dataset (zoom-in for a better view).** The shown examples are novel views reconstructed at unseen time instances. Note that in the flashlight scene, D-NeRF, NR-NeRF and T-NeRF fail to capture high-fidelity details in the background. On the other hand, in the cat-walking scene where the object moves across a considerable range in space, they fail to recover the moving object accurately. In the flower scene, where the motion is constrained within a small region, the baselines perform fairly well. In comparison, our method exhibits superior performance in all the cases.

Basis	Color Change			Falling and Scale			Light Move			Ball Move		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DCT	33.99	0.93	0.14	32.61	0.89	0.19	33.77	0.92	0.16	33.59	0.93	0.16
Fourier	31.33	0.89	0.19	29.74	0.89	0.21	31.99	0.91	0.23	33.45	0.94	0.19
Bernstein	27.81	0.86	0.21	28.90	0.87	0.25	31.57	0.91	0.25	33.53	0.94	0.18
Neural	**36.68**	**0.97**	**0.08**	**35.74**	**0.97**	**0.11**	**38.04**	**0.98**	**0.10**	**39.32**	**0.99**	**0.09**

Table 2. **Ablation of different time-basis functions.** Although other basis functions are able to yield acceptable performances, learned neural basis functions perform better.

5.3. Convergence

Our method converges around $20\times$ faster than D-NeRF and $10\times$ faster compared to NR-NeRF (Fig. 5). Also, we noticed that our convergence is more stable compared the baselines. For instance, NR-NeRF exhibited sudden divergences from the minima when the training is continued for a long time. Therefore, it was necessary to carefully monitor the training to determine the optimal termination point.

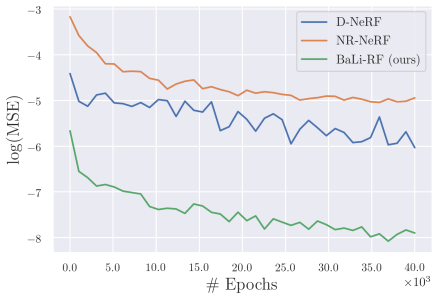

Figure 5. **Convergence.** Our model exhibits faster training compared to D-NeRF and NR-NeRF, and converges in $\sim 40k$ epochs. In comparison, D-NeRF and NR-NeRF take $\sim 800k$ and $\sim 200k$ epochs to converge, respectively. Time-wise, our model trains in ~ 1.5 hours per a scene, which is $\sim 20\times$ faster and $\sim 10\times$ faster compared to D-NeRF and NR-NeRF, respectively.

5.4. Ablation study

The generic nature of our framework allows different implementations. Thus, it is intriguing to compare the performance of other possible time-basis functions that are complete in $L^2(\mathbb{R}, dt)$ against neural trajectory basis. Table 2 presents a quantitative comparison with the DCT, Fourier, and Bernstein basis. As depicted, although these basis functions are also capable of providing acceptable results, neural trajectory basis performs best. This is a strong indicator of the effectiveness of the architectural regularization that is built into neural basis, which is vital in modeling complex dynamics. For further ablations refer to Supp. B.

6. Conclusions

We offer a novel, generic framework for modeling dynamic 3D scenes which allows efficient factorization of the space and time dynamics. This factorization presents a platform to impose well-designed space-time priors (inspired by NRS*f*M) on NeRF, enabling high-fidelity novel view synthesis of dynamics scenes. Finally, we present an implementation of the proposed framework that demonstrates compelling results across complex dynamics scenes containing long-range movements, scale changes, and light/texture changes.

References

[1] Antonio Agudo and Francesc Moreno-Noguer. Dust: Dual union of spatio-temporal subspaces for monocular multiple object 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6262–6270, 2017. 2, 5

[2] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. *Advances in neural information processing systems*, 21, 2008. 2, 16

[3] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1034–1040. IEEE, 1997. 1

[4] Shai Avidan and Amnon Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):348–357, 2000. 2

[5] Ronald T Azuma. A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4):355–385, 1997. 1

[6] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 690–696. IEEE, 2000. 2

[7] Grigore C Burdea and Philippe Coiffet. *Virtual reality technology*. John Wiley & Sons, 2003. 1

[8] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 1

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022. 4, 5

[10] Ismael Daribo and Béatrice Pesquet-Popescu. Depth-aided image inpainting for novel view synthesis. In *2010 IEEE International workshop on multimedia signal processing*, pages 167–170. IEEE, 2010. 1

[11] Mingsong Dou, Henry Fuchs, and Jan-Michael Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *2013 IEEE international symposium on mixed and augmented Reality (ISMAR)*, pages 99–106. Ieee, 2013. 2

[12] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 2

[13] Charles R Dyer. Volumetric scene reconstruction from multiple views. In *Foundations of image understanding*, pages 469–489. Springer, 2001. 1

[14] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 2, 3, 6

[15] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv preprint arXiv:2012.05903*, 2020. 2

[16] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 1272–1279, 2013. 2

[17] Paulo FU Gotardo and Aleix M Martinez. Kernel non-rigid structure from motion. In *2011 International Conference on Computer Vision*, pages 802–809. IEEE, 2011. 2

[18] Paulo FU Gotardo and Aleix M Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR 2011*, pages 3065–3072. IEEE, 2011. 2

[19] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010. 1

[20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[21] Erik Johnson, Marc Habermann, Soshi Shimada, Vladislav Golyanik, and Christian Theobalt. Unbiased 4d: Monocular 4d reconstruction with a neural deformation model. *arXiv preprint arXiv:2206.08368*, 2022. 2, 3

[22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 1

[23] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer, 2002. 1

[24] Chen Kong and Simon Lucey. Prior-less compressible structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4123–4131, 2016. 2

[25] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion. *Pattern Recognition*, 71:428–443, 2017. 2, 5

[26] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19, 2006. 16

[27] 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*, pages 6498–6508, 2021. 2, 3, 6

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2, 3

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 6

[30] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[31] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 2

[32] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 2

[33] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5379–5389, 2019. 2

[34] Hyun Soo Park and Yaser Sheikh. 3d reconstruction of a smooth articulated trajectory from a monocular image sequence. In *2011 International Conference on Computer Vision*, pages 201–208. IEEE, 2011. 2, 16

[35] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2, 3, 6, 16

[36] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2, 3, 6, 12, 16, 18, 25, 26, 27, 28, 29, 30

[37] Vincent Rabaud and Serge Belongie. Re-thinking non-rigid structure from motion. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2, 5

[38] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pages 142–158. Springer, 2022. 14

[39] Sameera Ramasinghe, Lachlan MacDonald, Moshiur Farazi, Hemanth Sartachandran, and Simon Lucey. How you start matters for generalization. *arXiv preprint arXiv:2206.08558*, 2022. 14

[40] Mathieu Salzmann and Raquel Urtasun. Physically-based motion models for 3d tracking: A convex formulation. In *2011 International Conference on Computer Vision*, pages 2064–2071. IEEE, 2011. 2, 16

[41] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1

[42] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1395, 2017. 2

[43] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Learning non-rigid 3d shape from 2d motion. *Advances in neural information processing systems*, 16, 2003. 2, 5

[44] Lorenzo Torresani, Danny B Yang, Eugene J Alexander, and Christoph Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001. 2, 5

[45] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 1, 2, 3, 6, 12, 13, 16

[46] Tony Tung, Shohei Nobuhara, and Takashi Matsuyama. Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1709–1716. IEEE, 2009. 2

[47] Jack Valmadre and Simon Lucey. General trajectory prior for non-rigid reconstruction. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1394–1401. IEEE, 2012. 2, 5, 16

[48] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 2, 3, 6

[49] Yonatan Wexler and Amnon Shashua. On the synthesis of dynamic scenes from reference views. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 576–581. IEEE, 2000. 2

[50] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 2, 3

[51] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *European Conference on Computer Vision*, pages 159–175. Springer, 2022. 2, 3

[52] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020. 2, 6

[53] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 910–919, 2017. 2

[54] Luca Zappella, Alessio Del Bue, Xavier Lladó, and Joaquim Salvi. Joint estimation of segmentation and structure from motion. *Computer Vision and Image Understanding*, 117(2):113–129, 2013. 2, 5

[55] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 1

[56] Li Zhang, Brian Curless, and Steven M Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–367. IEEE, 2003. 2

[57] Li Zhang, Noah Snavely, Brian Curless, and Steven M Seitz. Spacetime faces: high resolution capture for modeling and animation. In *ACM SIGGRAPH 2004 Papers*, pages 548–558. 2004. 2

[58] Yingying Zhu, Mark Cox, and Simon Lucey. 3d motion reconstruction for real-world camera motion. In *CVPR 2011*, pages 1–8. IEEE, 2011. 2, 16

[59] Yingying Zhu, Dong Huang, Fernando De La Torre, and Simon Lucey. Complex non-rigid motion 3d reconstruction by union of subspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1542–1549, 2014. 2, 5

[60] Yingying Zhu and Simon Lucey. Convolutional sparse coding for trajectory reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):529–540, 2013. 2

Supplementary Materials

A. Ray Deformation Networks

A.1. Ray deformation networks learn field dynamics

In this section, we show evidence that the design methodology adopted by existing works for implementing the ray deformation framework do not learn point trajectories in space, and instead, act as light and density deformation modules. Consider an MLP $\Psi^d : (x, y, z, t) \rightarrow (\Delta x, \Delta y, \Delta z)$ outputting the transformation of a point (x, y, z) at time instant t with respect to a canonical setting. Then, a second MLP $\Psi^c : (x + \Delta x, y + \Delta y, z + \Delta z, \mathbf{d}) \rightarrow (c, \sigma)$ takes in the deformed inputs and the viewing direction $\mathbf{d}$, and predicts the density and light fields (c, σ). However, one can interpret the above pipeline from another perspective. Observe that Ψ^d and Ψ^c can be considered as a single deep MLP $\Psi^{d \wedge c} : (x, y, z, t) \rightarrow (c, \sigma)$, where the bottleneck is three-dimensional. Further, there exists a skip connection from (x, y, z) to the bottleneck. From this perspective, the above implementation is simply an MLP with a skip connection and bottleneck of dimension three, modeling a function from (x, y, z, t) to (c, σ). See Fig. 8 for a visual illustration of this interpretation. We empirically solidify this argument by showing that such networks can indeed model light and density deformations individually (to an extent), which is impossible with a model that only learns point movements in space, according to the ray deformation framework (see Fig. 7). Next, we discuss limiations of ray deformation networks.

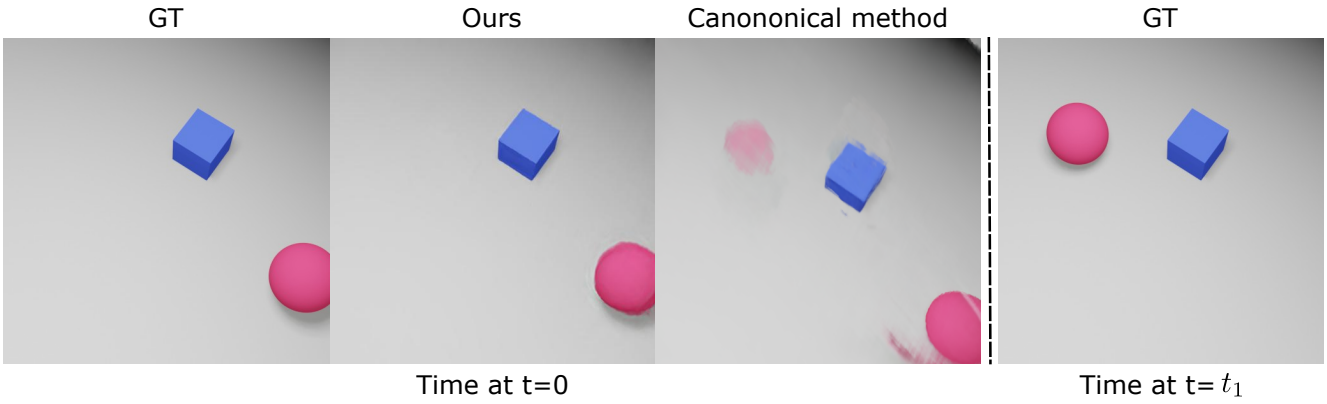

Figure 6. **Our method does not rely on a canonical scene configuration**. Existing ray deformation models require choosing a canonical scene configuration at a user-defined time instance, which can hinder their performance in scenes where new information appears in subsequent frames. In contrast, our model does not suffer from such a limitation.

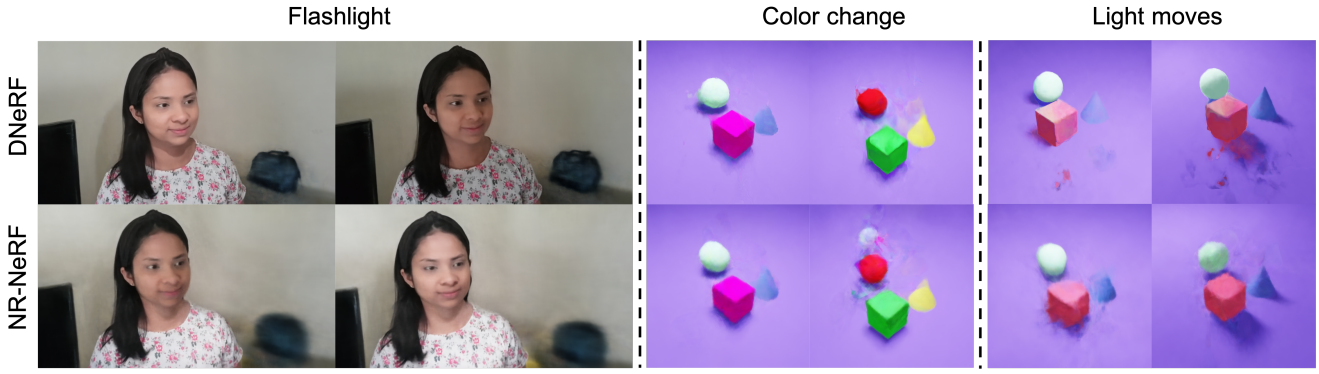

Figure 7. **Ray deformation networks parameterize light and density fields instead of ray deformations.** We show evidence for this using three example scenes. From left to right, *1)* A real world scene with light changes on the person's face. *2)* The colors of the shapes are changing. *3)* The light is shifting position in the scene. We can observe both ray deformation works, DNeRF [36] and NR-NeRF [45], learn the texture changes that occur to an extent, which is not possible by simply learning ray deformations. However, the reconstructions are still sub-par as light and density dynamics are entangled in these frameworks.

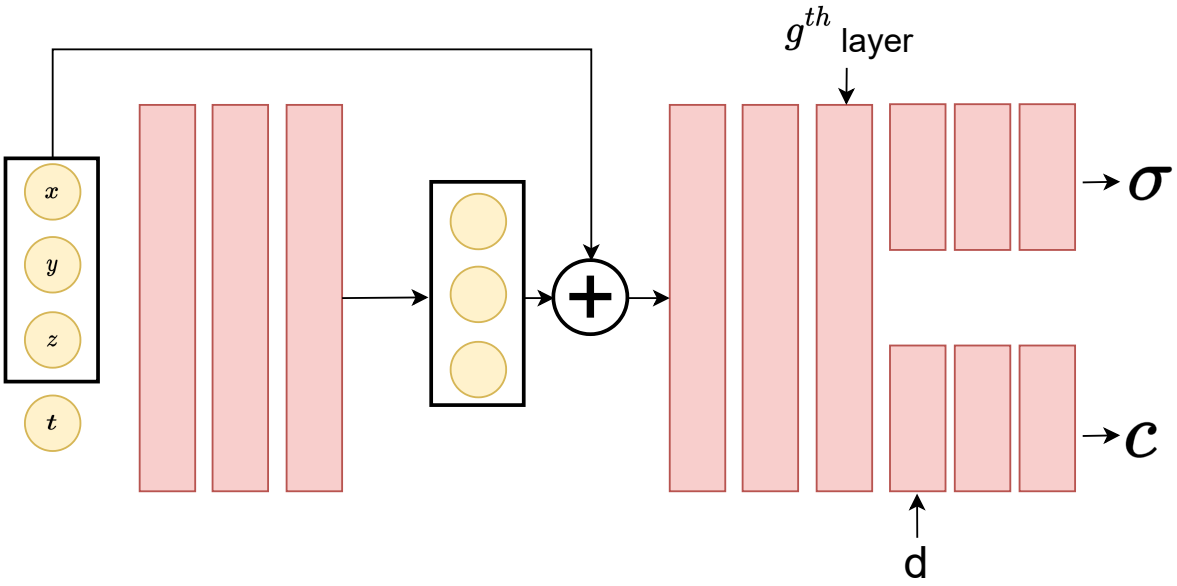

Figure 8. **Ray deformation networks can be interpreted as a single deep network with a three-dimensional bottleneck.** With this interpretation, it is clear that ray deformation networks can indeed learn light and density evolution independently (to an extent), instead of simply learning ray deformations.

A.2. Dependency on a canonical frame

A weakness that comes with learning a canonical frame is one root cause for ray deformation networks to struggle in scenes with an object that has long translations. This type of formulation is also working against the smoothness assumptions of neural networks, which are now required to learn non-smooth representations. In Fig. 6 we can see a toy example of a ball moving in a constant trajectory across the scene. The canonical method at time step $t = 0$ is forced to learn an average representation of the scene and is unable to correctly represent the canonical frame which corresponds to image GT at time $t = 0$. This formulation also negatively impacts ray deformation models to encode fine detail information, due to the constant averaging the canonical frame maintains throughout time. In Fig. 1 it can be seen that NR-NeRF [45] fails to capture fine details such as shirt wrinkles and the climbers legs.

A.3. Entanglement of light and density fields

Let the output of the g^{th} layer of Fig. 8 be $g(\mathbf{x}) : \mathbb{R}^4 \rightarrow \mathbb{R}^C$. Further, let $\psi_l : \mathbb{R}^C \rightarrow \mathbb{R}$ and $\psi_d : \mathbb{R}^C \rightarrow \mathbb{R}$ be network branches that predict light and density fields, respectively, taking $g(\mathbf{x})$ as input. Now, consider a scenario where the light of the scene or the texture of objects change, while the objects remain static. In this case, we need the light field to be a function of time, while the density field should remain constant. Consider the Jacobians,

$$\mathbf{J}_g = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x} & \frac{\partial g_1(\mathbf{x})}{\partial y} & \frac{\partial g_1(\mathbf{x})}{\partial z} & \frac{\partial g_1(\mathbf{x})}{\partial t} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_C(\mathbf{x})}{\partial x} & \frac{\partial g_C(\mathbf{x})}{\partial y} & \frac{\partial g_C(\mathbf{x})}{\partial z} & \frac{\partial g_C(\mathbf{x})}{\partial t} \end{bmatrix}, \tag{13}$$

$$\mathbf{J}_{\psi_d} = \begin{bmatrix} \frac{\partial \psi_d(g(x))}{\partial g_1(\mathbf{x})} & \frac{\partial \psi_d(g(x))}{\partial g_2(\mathbf{x})} & \dots & \frac{\partial \psi_d(g(x))}{\partial g_C(\mathbf{x})}. \end{bmatrix} \tag{14}$$

Then, the Jacobian of $\psi_d \circ g$ becomes,

$$\mathbf{J}_{\psi_d \circ g} = \begin{bmatrix} \frac{\partial \psi_d(g(\mathbf{x}))}{\partial x} & \frac{\partial \psi_d(g(\mathbf{x}))}{\partial y} & \frac{\partial \psi_d(g(\mathbf{x}))}{\partial z} & \frac{\partial \psi_d(g(\mathbf{x}))}{\partial t} \end{bmatrix} = \mathbf{J}_{\psi_d} \mathbf{J}_g. \tag{15}$$

And, we need $\frac{\partial \psi_d(g(x))}{\partial t} = 0$ since the density is not a function of time. Therefore, the 4^{th} column of $\mathbf{J}_g$ has to be orthogonal to $\mathbf{J}_{\psi_d}$. This can be achieved via one of the following three scenarios:

- **Scenario 1:** $\mathbf{J}_{\psi_d}$ *is a zero vector.*
- **Scenario 2:** *The* 4^{th} *column of* $\mathbf{J}_g$ *is zero.*
- **Scenario 3:** *Both scenario 1 and 2 are false, but* $\mathbf{J}_{\psi_d}$ *is orthogonal to the* 4^{th} *column of* $\mathbf{J}_g$.

However, Scenario 1 implies that $\frac{\partial \psi_d \circ g(\mathbf{x})}{\partial x,y,z}$ is zero (see Eq. 15), which makes the density constant across space. On the other hand, Scenario 2 implies that $\frac{\partial \psi_l}{\partial t}$ is zero since $\frac{\partial \psi_l}{\partial t} = \frac{\partial \psi_l}{\partial g}\frac{\partial g}{\partial t}$. That is, with Scenario 2, the light cannot be a function of time. Further, in general, Scenario 3 makes $\mathbf{J}_{\psi_d}$ a function of $\frac{\partial g(\mathbf{x})}{\partial t}$ since in the case where g obeys the following PDE,

$$\frac{\partial g(\mathbf{x})}{\partial t} = q(t), \tag{16}$$

where $q(t)$ is some function parameterized by t. Thus, it is clear that $\mathbf{J}_{\psi_d}$ becomes a function of t. On the other hand, by Eq. 15, $\frac{\partial \psi_d \circ g(\mathbf{x})}{\partial x,y,z}$ also becomes a function of time, unless both $\mathbf{J}_{\psi_d}$ and $\mathbf{J}_g$ preserves a block structure such that

$$\mathbf{J}_g = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x} & \frac{\partial g_1(\mathbf{x})}{\partial y} & \frac{\partial g_1(\mathbf{x})}{\partial z} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_c(\mathbf{x})}{\partial x} & \frac{\partial g_c(\mathbf{x})}{\partial y} & \frac{\partial g_c(\mathbf{x})}{\partial z} & 0 \\ 0 & 0 & 0 & \frac{\partial g_{c+1}(\mathbf{x})}{\partial t} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \frac{\partial g_C(\mathbf{x})}{\partial t} \end{bmatrix}, \tag{17}$$

and

$$\mathbf{J}_{\psi_d} = \begin{bmatrix} \frac{\partial \psi_d(gx)}{\partial g_1(\mathbf{x})} & \dots & \frac{\partial \psi_d(gx)}{\partial g_c(\mathbf{x})} & 0 & \dots & 0 \end{bmatrix} \tag{18}$$

Note that this is an extremely unique solution that is seldom achieved in practice under general conditions, due to the ill-posed nature of the problem. In most cases, the networks tend to converge to solutions where the 4^{th} column of $\mathbf{J}_g$ becomes non-zero, in order to model the light changes, which in turn makes the density a function of time. This causes an inherent entanglement of the light and density fields. The toy example results shown in Fig. 7 is an illustration of this behavior.

A.4. Limited expressiveness

Consider the parameterization of the density field. As evident from Fig. 8, it is modeled with a network with a bottleneck of dimension three. In this setting, the density field becomes a manifold of dimension three. In other words, the dynamics of the density field can be modeled with only three parameters. However, recall that in complex scenes, particular points of the density field may need to be parameterized by (x, y, z, t) simultaneously. In other words, it is required that $\frac{\partial \sigma}{\partial x}, \frac{\partial \sigma}{\partial y}, \frac{\partial \sigma}{\partial z}, \frac{\partial \sigma}{\partial t} \neq 0$ at some points in space-time. Thus, having a bottleneck of three hinders the network from modeling such complex dynamics.

A.5. Entanglement of space and temporal variations

A ray deformation network can be considered as a single deep network with bottleneck three, that takes (x, y, z, t) as inputs and models light/density field deformations. However, often, space and time consist of contrasting spectral properties; objects deform smoothly across time, but space may contain sharp/high frequency variations. Therefore, using a single neural network to model this two extremes can be sub-optimal. Generally, a network with a higher bandwidth is ideal for modeling space, and a lower bandwidth is necessary for modeling time.

Fig. 9 shows an illustration. Using a high bandwidth network for interpolating across time allows a network to perfectly memorize training data, but can result in erratic interpolations. On the other hand, using a low-bandwidth network leads to low fidelity space reconstructions. Note that since space is typically more densely sampled (compared to time axis) in the dynamic NeRF setting, a high-bandwidth network can recover both low and high frequencies in space, i.e., supervision is available more densely. This is a key factor that motivates space/time factorization, as in our framework. This behavior was also observed previously by [38] and [39].

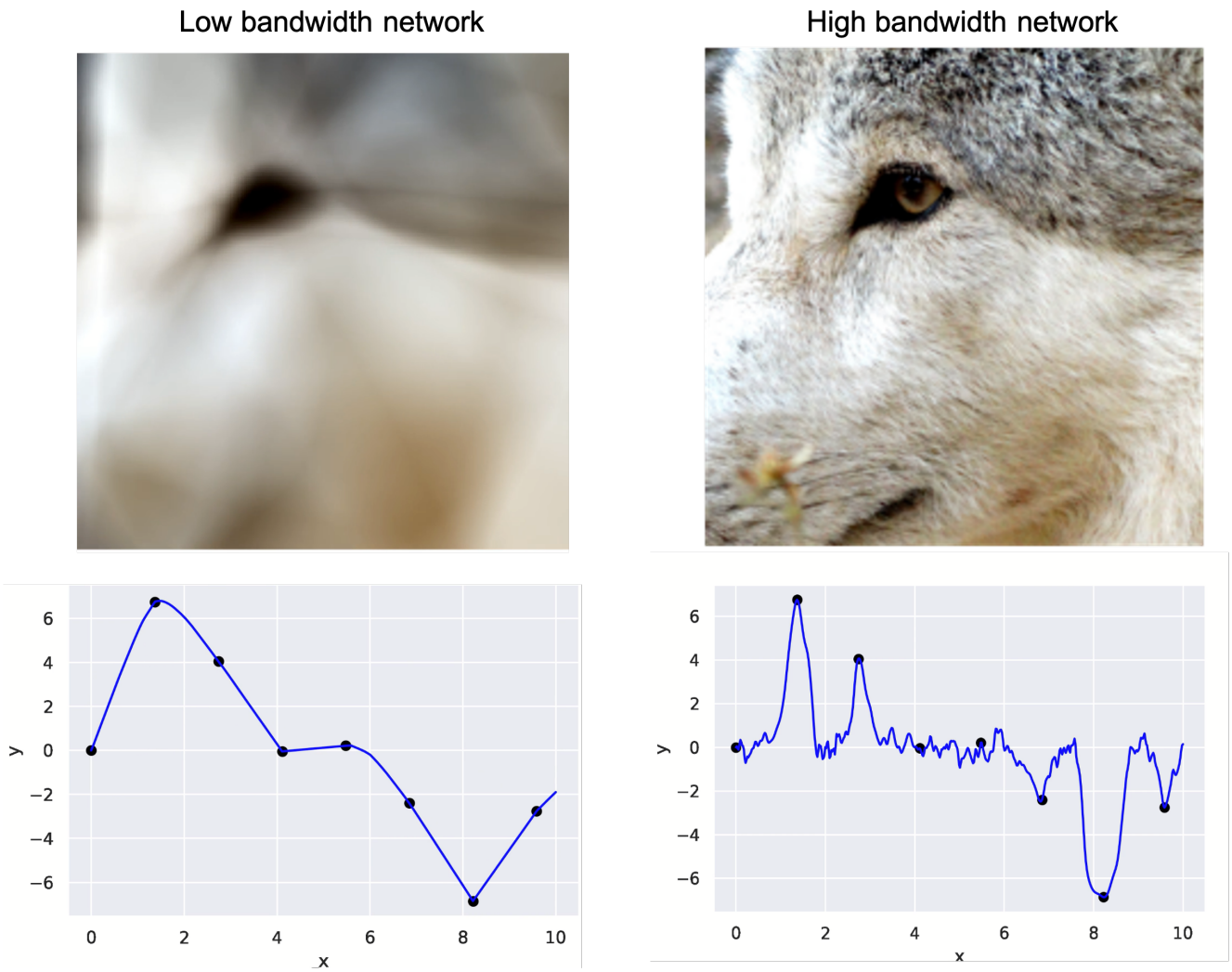

Figure 9. *Left column:* A low-bandwidth network cannot capture high-frequency content adequately, which can be sub-optimal for modeling space. However, a low-bandwidth network is ideal for interpolating sparse low frequency points, which is optimal for modeling temporal dynamics. *Right column:* A high-bandwidth network can reconstruct sharp variations, but results in erratic interpolations. This can be detrimental for smooth temporal dynamics modeling. We used four layer ReLU networks with positional embeddings for encoding the signals. We obtained networks with different bandwidths by changing the frequency support of the positional embedding layer.

B. Ablations

In this section, we show experiments over varying number of basis functions and the effect of manifold regularization. As seen in Table. 3, the performance saturates at around 24 basis functions. Further, the effect of manifold regularization is quite significant (Table. 4).

# Basis	Color Change		Falling and Scale		Light Move		Ball Move	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
4	33.16	0.96	12.04	0.81	13.44	0.81	13.59	0.82
12	35.19	0.95	28.18	0.89	27.50	0.88	26.91	0.88
24	36.68	0.97	35.74	0.97	38.04	0.98	39.32	0.99
48	36.61	0.97	35.19	0.97	38.04	0.98	39.33	0.99

Table 3. Performance against the number of time-basis functions.

# Regularization	Color Change		Falling and Scale		Light Move		Ball Move	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
W/O manifold regularization	33.11	0.96	32.16	0.96	38.00	0.98	36.17	0.97
W/manifold regularization	36.68	0.97	35.74	0.97	38.04	0.98	39.32	0.99

Table 4. The effect of manifold regularization.

C. Implicit regularization of the neural trajectory basis

Using a combination of trajectory basis to reconstruct the motion of a set of points is popular in NRS*f*M [2]. This technique implicitly restricts the solution to a known low-dimensional subspace of smooth trajectories. One such popular trajectory basis is the DCT basis. A key advantage of this method compared to a shape basis is that an object-agnostic basis can be employed across multiple scenes. However, although the basis type is scene agnostic, the basis dimensionality depends on multiple factors such as scene dynamics, camera dynamics, and sequence length [34]. Thus, the dimensionality of the basis functions should be tuned per scene.

In an attempt to solve the above problem, [58] applied an ℓ_1 norm penalty on the coefficients of the trajectory basis. In practice, a sparse-coding algorithm [26] was used to achieve this. Although this strategy was effective, it ignores an important prior; for natural signals, the DCT basis tends to concentrate of the lower frequencies.

An alternative and a more effective way of regularizing the trajectory basis has been minimizing the trajectory responses to high-pass filters. [47] showed that such regularization is able to enforce local temporal constraints, rather than global constraints, which extends trivially to sequences of different length. They particularly showed that this mechanism alleviate the need to tune the basis size. This approach also has a physical interpretation; minimizing the ℓ_2 norm of the second-order derivative is equivalent to an assumption of constant mass subject to isotropic Gaussian distributed forces [40]. Similarly, minimizing the ℓ_2 norm of the first-order derivative is equivalent to finding the solution with the least kinetic energy.

In our architecture also we observed similar behaviors. As shown in the blue curve of the left figure in Fig. 10, when the number of basis functions is increased, the loss reaches a minimum, but then increases again. This aligns with the intuition that the motion should be restricted to a low-dimensional manifold of smooth trajectories. Then, we apply 1D convolutions on the DCT trajectories with kernels $[-1, 1]$ and $[-1, 2, -1]$, and minimize the ℓ_1 norm on the convolution outputs. These DCT trajectories are then used as the basis functions for modeling the light and density field temporal dynamics. As shown by the orange curve, with this strategy, the performance of the model becomes almost agnostic to the basis size after the minimum. This result is similar to the conclusions of [47].

Interestingly, we observed that with the neural basis, this regularization is implicitly achieved; see the right plot of Fig. 10. The shown results are for the test set of the ball moves scene. As evident, the performance of the model almost saturates after a certain number of basis functions, eliminating the need for carefully tuning the number of basis functions for each sequence. This result is a powerful indication of the strong architectural bias that stems from the neural networks.

D. Datasets and evaluation

We collect four synthetic scenes and four real-world scenes as our dataset. All the scenes consist of RGB images captured from a single moving camera along with camera poses. The synthetic scenes are color change, falling and scale, light move, and ball move. The color change scene includes texture changes of objects. The light move scene contains static objects, but a moving light source. The falling and scale scene contains objects that change scale, and the ball move scene consist of objects with long-range movements. Similarly, the real world scenes are climbing, cat walking, flashlight, and flower. The climbing and cat walking scenes contain long-range movements, while the flashlight scene contains light changes. In contrast, the flower scene contains spatially concentrated dynamics.

For each the real world scene, we used 12 consecutive frames as training frames, and the subsequent 4 frames as testing frames, throughout the video. For the synthetic scenes, we used an unseen fixed pose to render the test frames across time. For evaluation, we used PSNR, SSIM, and LPIPS, as commonly done in literature [35, 36, 45].

E. Hyperparameters and training

We use 24 basis functions for modeling each of the light and density fields. For manifold regularization, we use 8 as the submanifold dimension. For generating the neural trajectories, we use three-layer ReLU networks with positional

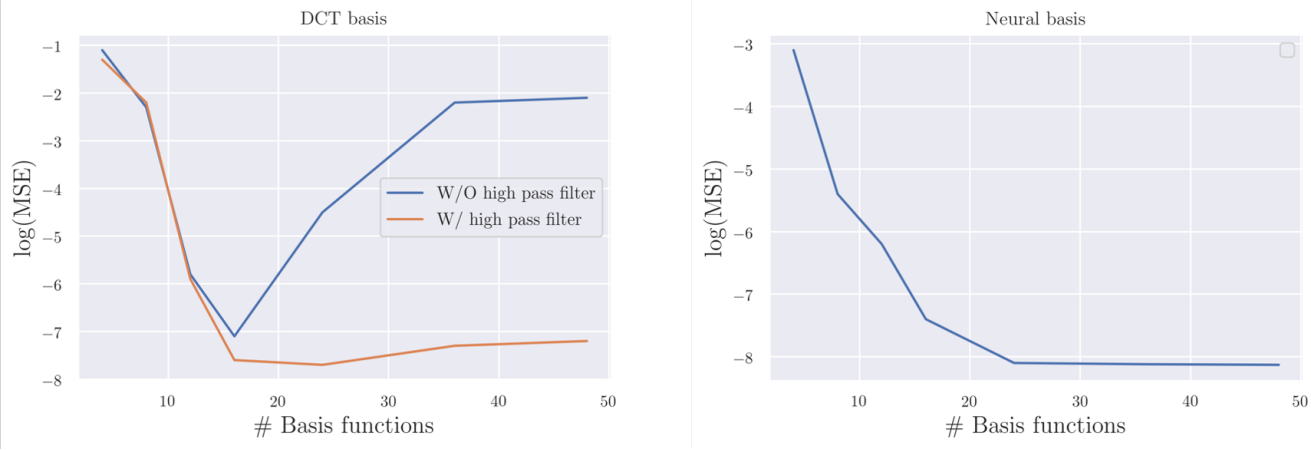

Figure 10. **Implicit regularization of the neural trajectory basis.** *Left:* With the DCT basis, the performance is sensitive to the number of basis functions. After an optimal basis size, the performance decreases. However, this can be avoided with penalizing the trajectory output on convolutional kernels ($[-1, 1], [-1, 2, -1]$). *Right:* This regularization is implicitly achieved by the neural basis. After a certain number of basis functions, the performance remains approximately the same.

embeddings. We choose 0.1 for λ_1 and λ_2 in Eq. 12. For training, we used an ADAM optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We used cyclic learning rates for training both neural networks and the coefficient tensors. For the neural networks, we start the learning rate at 0.001, and for the coefficient tensors, we start the learning rate at 0.02.

F. T-TensoRF

Two unique features of our framework are the light/density disentanglement and the space/time factorization. Thus, it is necessary to properly evaluate the superior performance of our model against these two factors. To this end, we design a baseline which completely disentangles the light and density fields, but does not factorize space and time. Fig. 11 shows the overall architecture. Here, we first model the light $\mathcal{S}_c$ and density $\mathcal{S}_\sigma$ fields as 3D tensors, which is decomposed in to a linear combination of outer products between matrices and vectors:

$$\mathcal{S}_\sigma = \sum_{j=1}^{N} (\mathbf{v}^z_{\sigma,j} \otimes \mathbf{M}^{xy}_{\sigma,j} + \mathbf{v}^x_{\sigma,j} \otimes \mathbf{M}^{yz}_{\sigma,j} + \mathbf{v}^y_{\sigma,j} \otimes \mathbf{M}^{xz}_{\sigma,j}), \tag{19}$$

$$\mathcal{S}_c = \sum_{j=1}^{N} (\mathbf{v}^z_{c,j} \otimes \mathbf{M}^{xy}_{c,j} + \mathbf{v}^x_{c,j} \otimes \mathbf{M}^{yz}_{c,j} + \mathbf{v}^y_{c,j} \otimes \mathbf{M}^{xz}_{c,j}), \tag{20}$$

For querying continuous 3D positions, we tri-linearly interpolate the resultant grid. Let

$$R_{c,j} = (\mathbf{v}^z_{c,j} \otimes \mathbf{M}^{xy}_{c,j} + \mathbf{v}^x_{c,j} \otimes \mathbf{M}^{yz}_{c,j} + \mathbf{v}^y_{c,j} \otimes \mathbf{M}^{xz}_{c,j}) \tag{21}$$

and

$$R_{\sigma,j} = (\mathbf{v}^z_{\sigma,j} \otimes \mathbf{M}^{xy}_{\sigma,j} + \mathbf{v}^x_{\sigma,j} \otimes \mathbf{M}^{yz}_{\sigma,j} + \mathbf{v}^y_{\sigma,j} \otimes \mathbf{M}^{xz}_{\sigma,j}), \tag{22}$$

and $R_{c,j}(\mathbf{x})$, $R_{\sigma,j}(\mathbf{x})$ denote the values queried at $\mathbf{x}$. Then, we use two linear networks $L_\sigma, L_c : \mathbb{R}^N \rightarrow \mathbb{R}^F$ to generate F-dimensional density/light feature vectors (μ_σ, μ_c) for each 3D position $\mathbf{x}$ as

$$\mu_\sigma = L_\sigma(R_{\sigma,1}, \dots R_{\sigma,N}), \tag{23}$$

$$\mu_c = L_c(R_{c,1}, \dots R_{c,N}). \tag{24}$$

This is equivalent to generating a density/light feature vector for each 3D position of the scene. Then, we concatenate these feature vectors with the scalar time value, and feed to a 4-layer ReLU network to obtain color and density values for each $\mathbf{x}$. The neural rendering and training is done similar to our model.

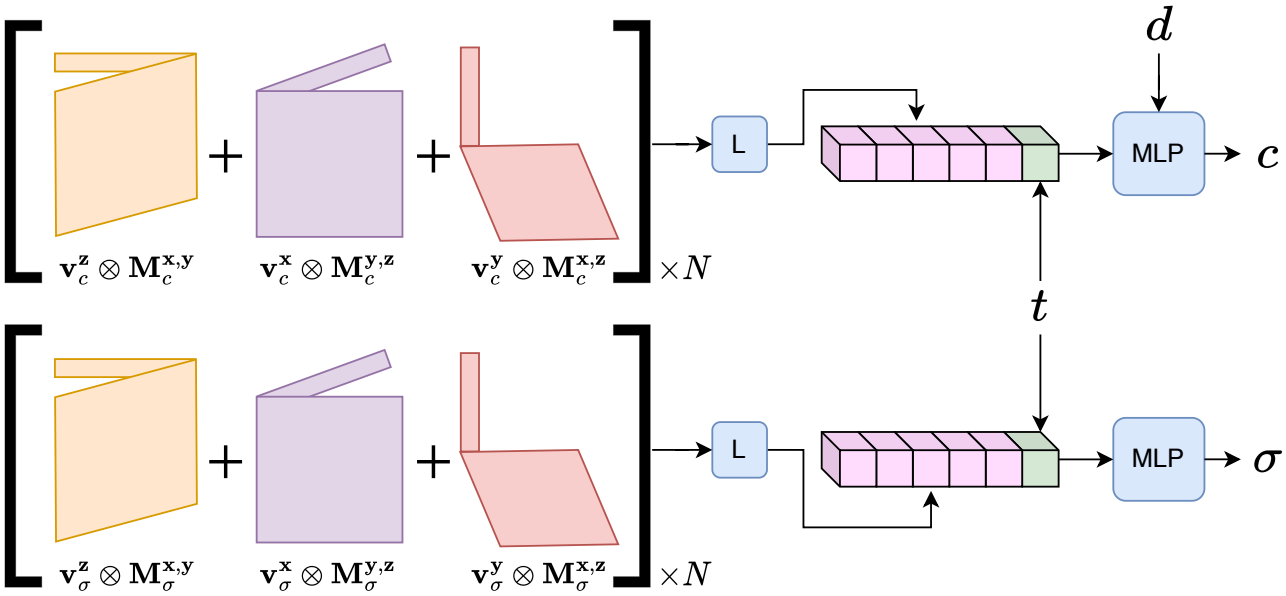

Figure 11. **The T-TensoRF architecture.** We develop a baseline that disentangles light and density fields, but does not factorize time and space. See Sec. F for a detailed description.

G. Novel view generation

In this section, we offer more qualitative comparisons. For real world scenes, we first fix the pose and move time to generate novel views. Fig. 12, 13, 14, and 15 depict results. Next, we fix the time and generate novel views by changing the poses. Fig. 16, 18, 17, and 19 depict corresponding results. As evident, our model exhibits significantly superior performance over all the instances. Recall that the training images for these scenes are obtained from a single moving camera. Therefore, only a single image is available for a particular time instance. Thus, this reconstruction task is a severely underconstrained problem, specially in the context of complex real world dynamics. Therefore, the superior results shown by our model is a strong indicator of its inbuilt architectural bias that implicitly regularizes the problem.

We also conduct experiments over the synthetic dataset released by [36]. Fig. 20, 21, 23, 24, 22, and 25 depict results. As shown, our model is able to generate novel views in both constant pose and constant time settings.

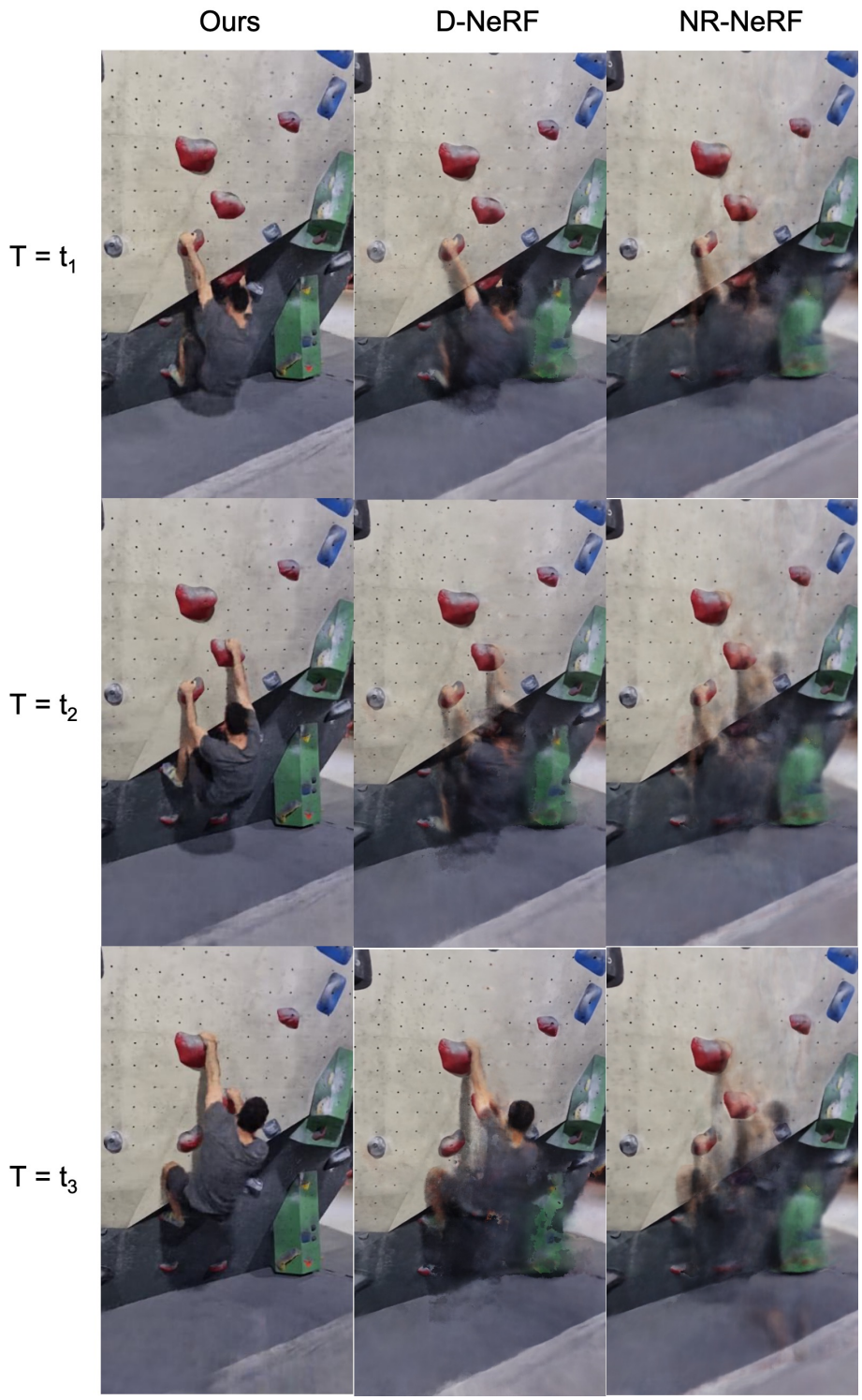

Figure 12. **A qualitative comparison over the generated novel views on the climbing scene.** We fix the pose and generate views by varying time. As depicted, our model is able to achieve superiors results in all the instances.

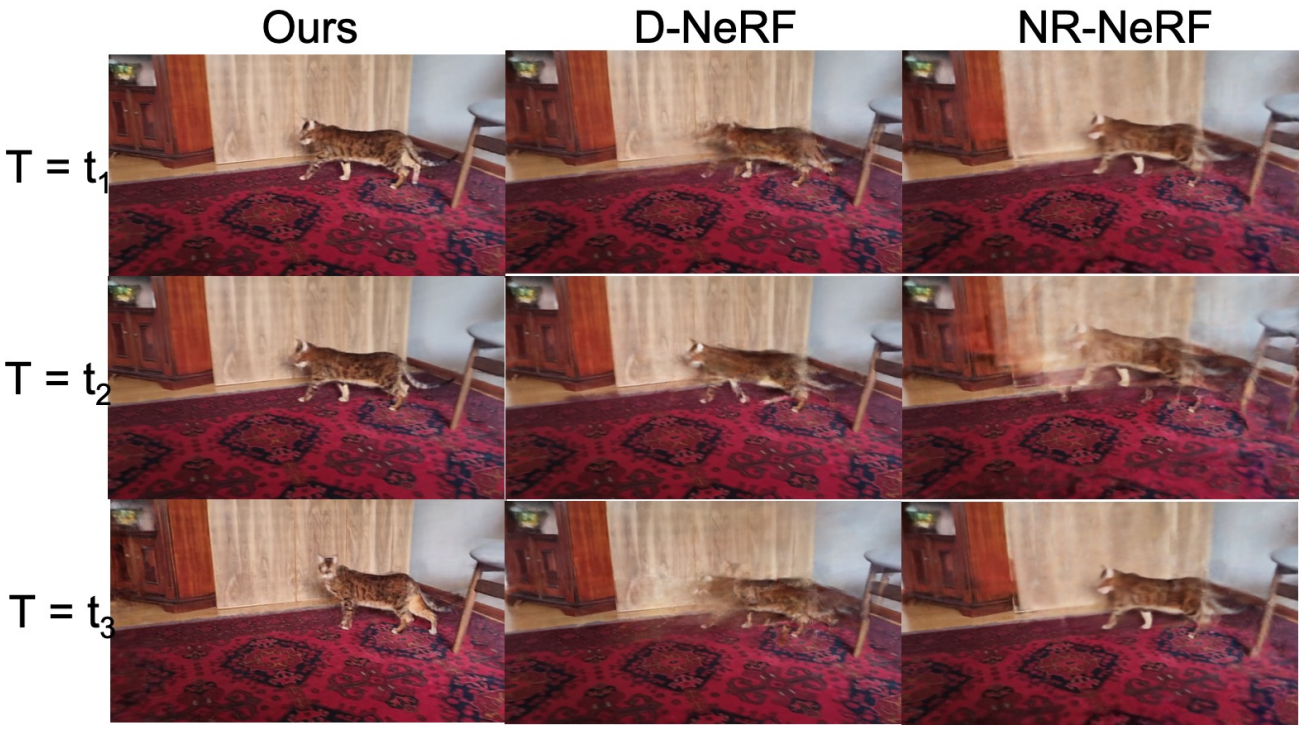

Figure 13. **A qualitative comparison over the generated novel views on the cat scene.** We fix the pose and generate views by varying time. As depicted, our model is able to achieve superiors results in all the instances.

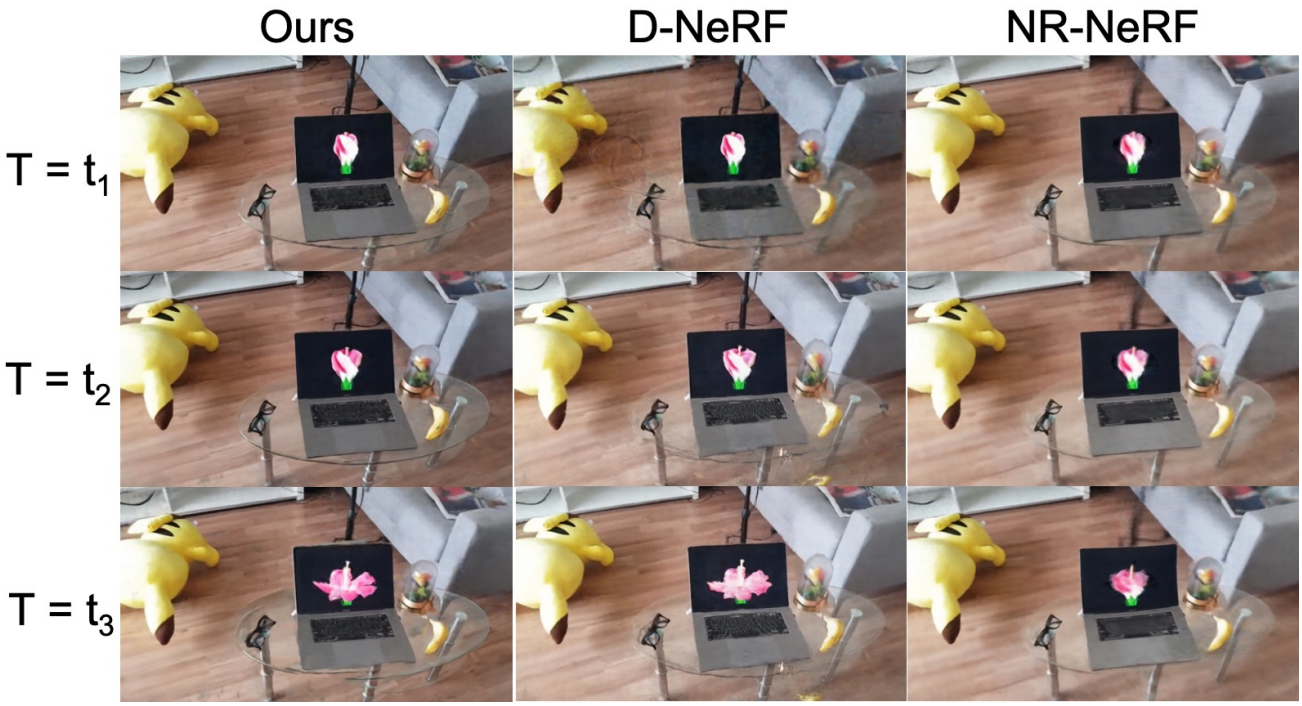

Figure 14. **A qualitative comparison over the generated novel views on the flower scene.** We fix the pose and generate views by varying time. As depicted, our model is able to achieve superiors results in all the instances.

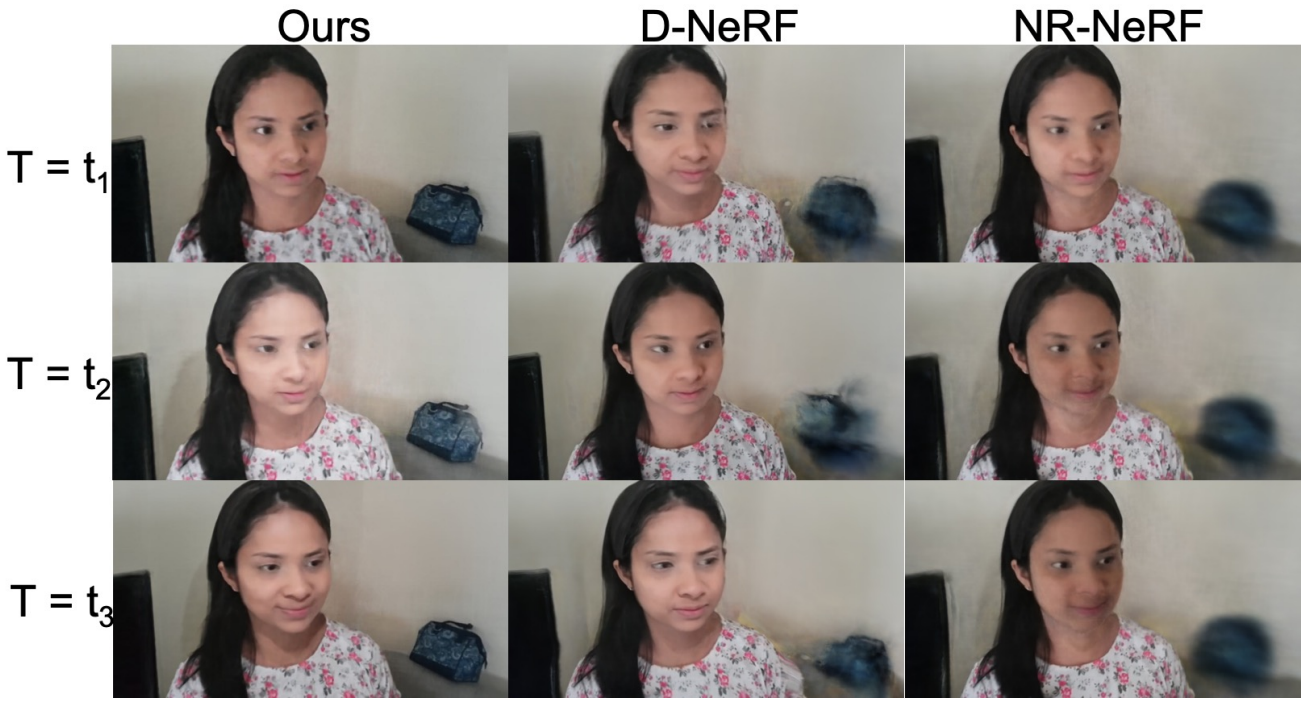

Figure 15. **A qualitative comparison over the generated novel views on the flashlight scene.** We fix the pose and generate views by varying time. As depicted, our model is able to achieve superiors results in all the instances.

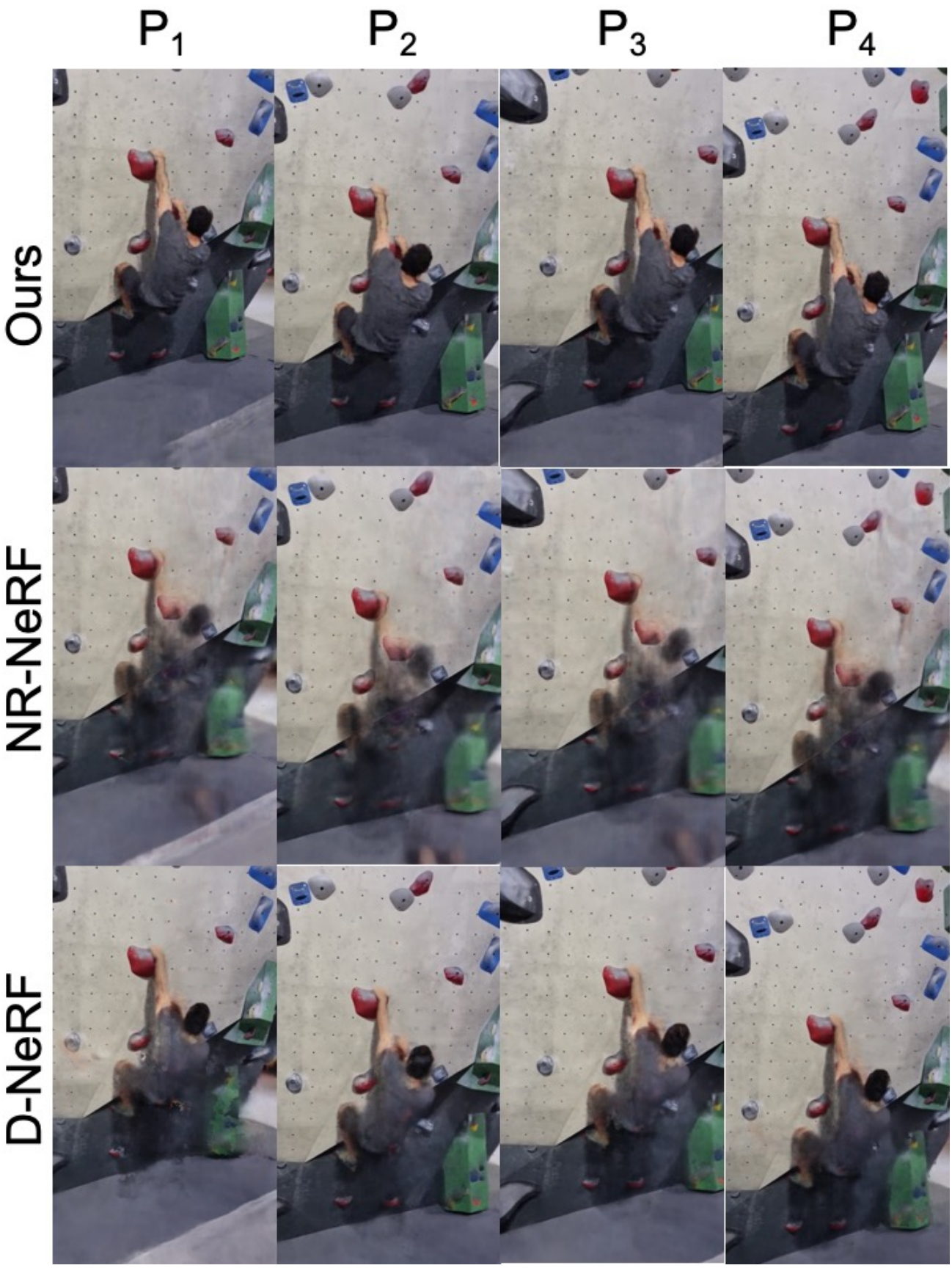

Figure 16. **A qualitative comparison over the generated novel views on the climbing scene.** We fix the time and generate views from different camera poses. As depicted, our model is able to achieve superiors results in all the instances.

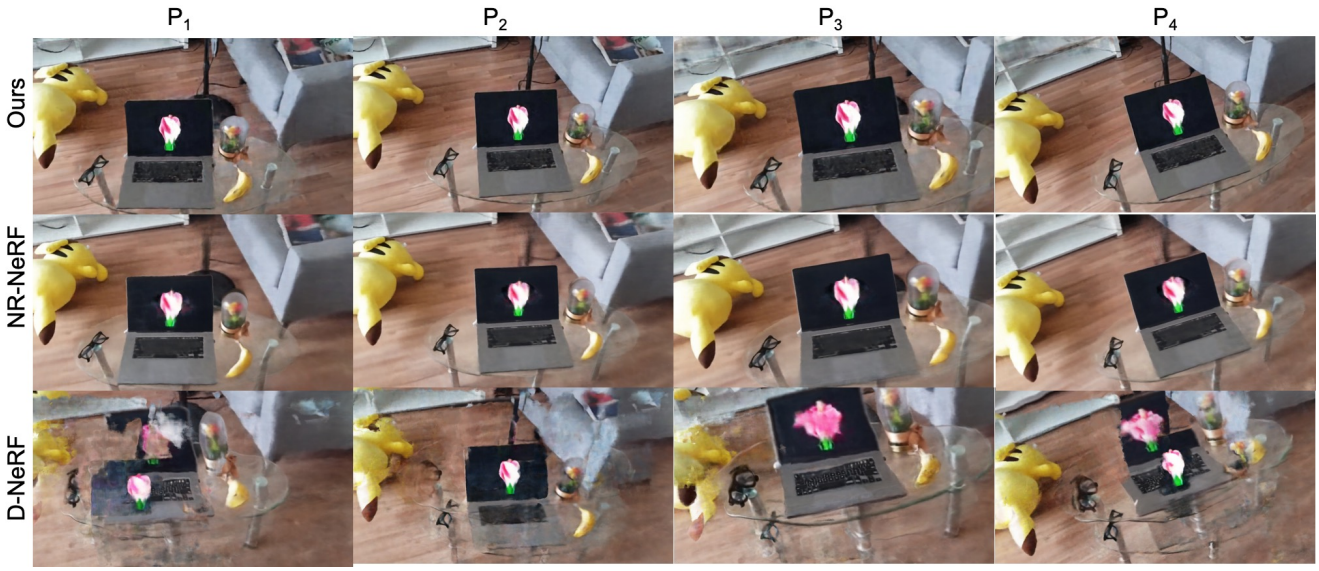

Figure 17. **A qualitative comparison over the generated novel views on the flower scene.** We fix the time and generate views from different camera poses. As depicted, our model is able to achieve superiors results in all the instances.

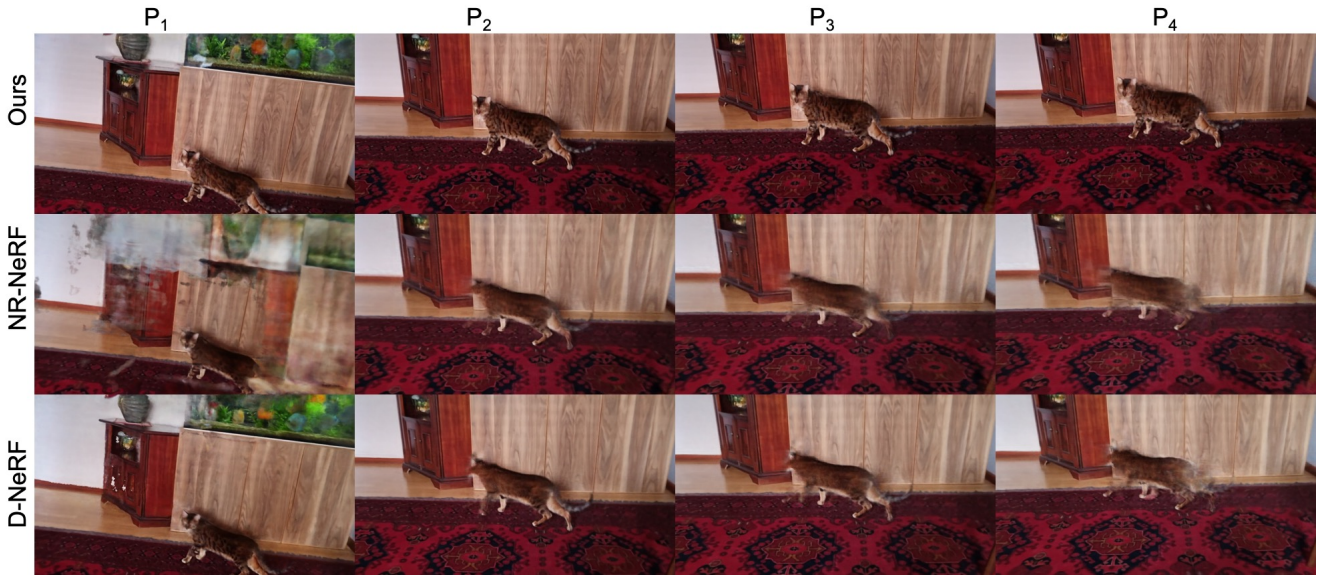

Figure 18. **A qualitative comparison over the generated novel views on the cat scene.** We fix the time and generate views from different camera poses. As depicted, our model is able to achieve superiors results in all the instances.

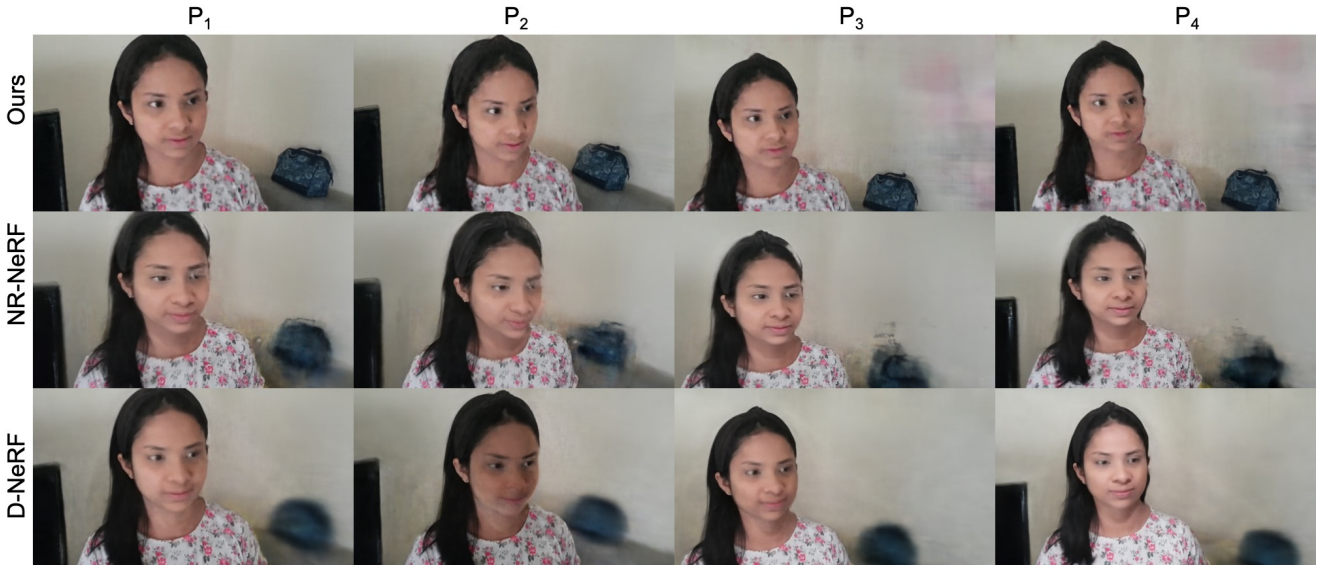

Figure 19. **A qualitative comparison over the generated novel views on the flashlight scene.** We fix the time and generate views from different camera poses. As depicted, our model is able to achieve superiors results in all the instances.

Figure 20. Qualitative examples generated by our model on the *standup* scene in [36] synthetic dataset.

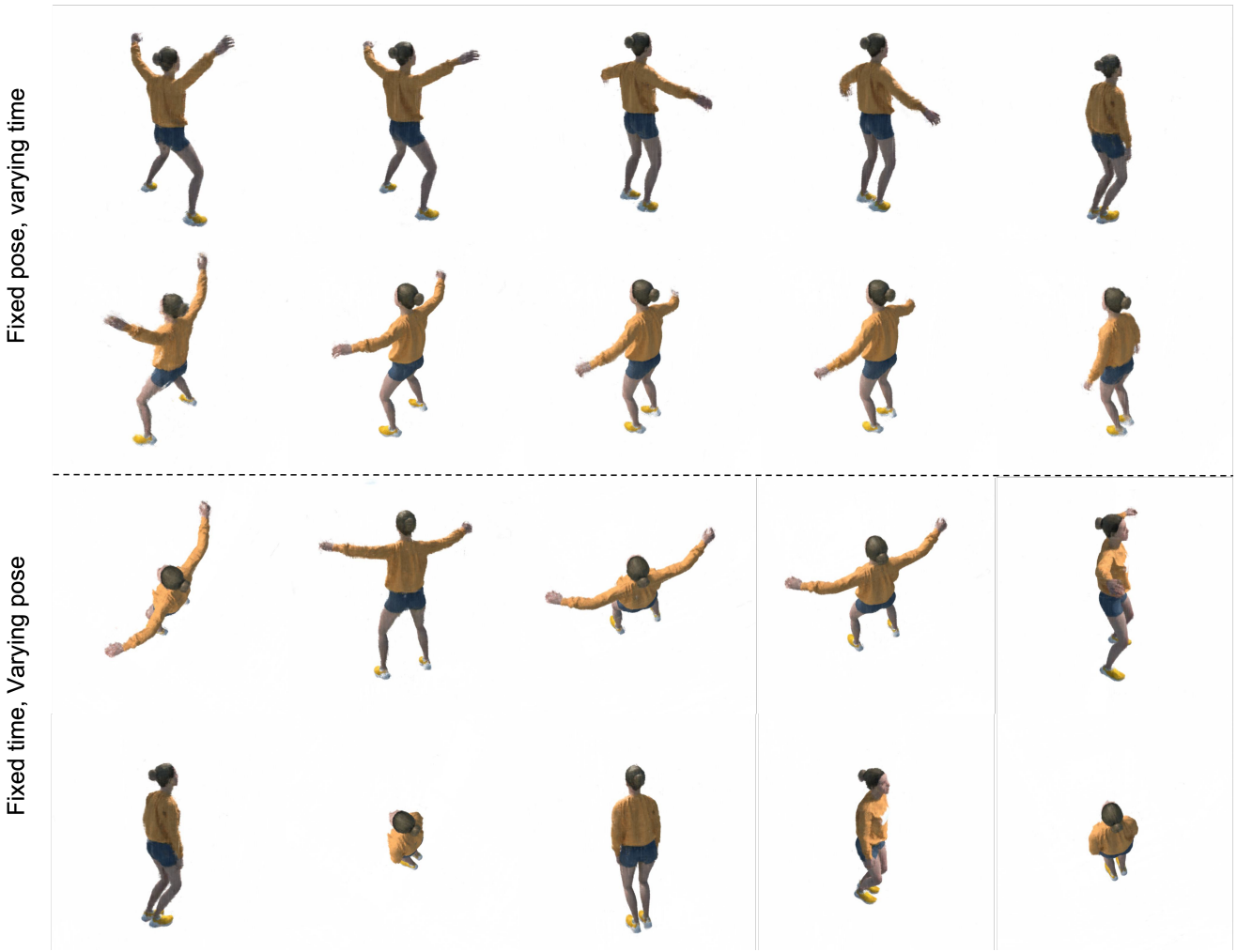

Figure 21. Qualitative examples generated by our model on the *jumping* scene in [36] synthetic dataset.

Figure 22. Qualitative examples generated by our model on the *trex* scene in [36] synthetic dataset.

Figure 23. Qualitative examples generated by our model on the *mutant* scene in [36] synthetic dataset.

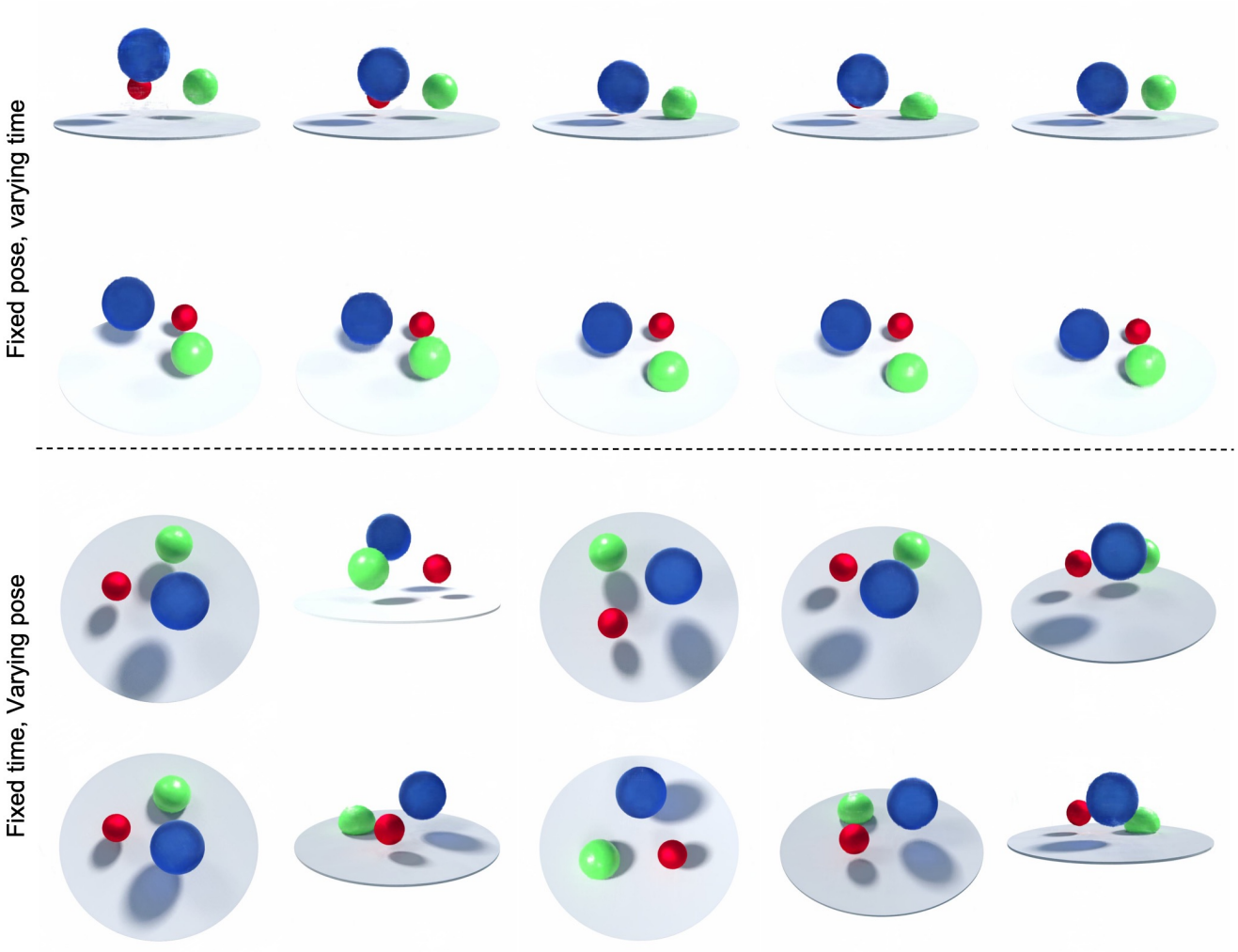

Figure 24. Qualitative examples generated by our model on the *bouncing balls* scene in [36] synthetic dataset.

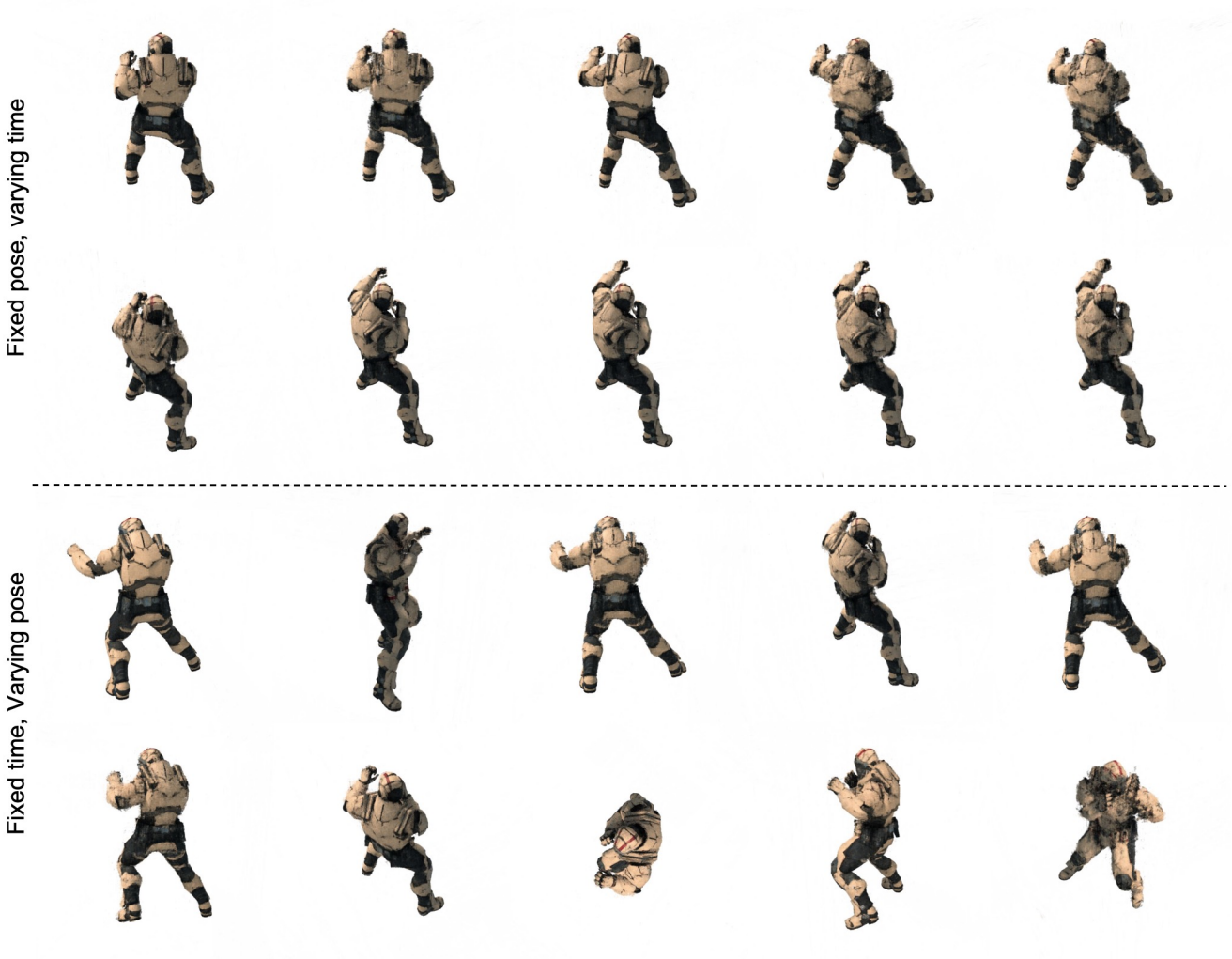

Figure 25. Qualitative examples generated by our model on the *hook* scene in [36] synthetic dataset.