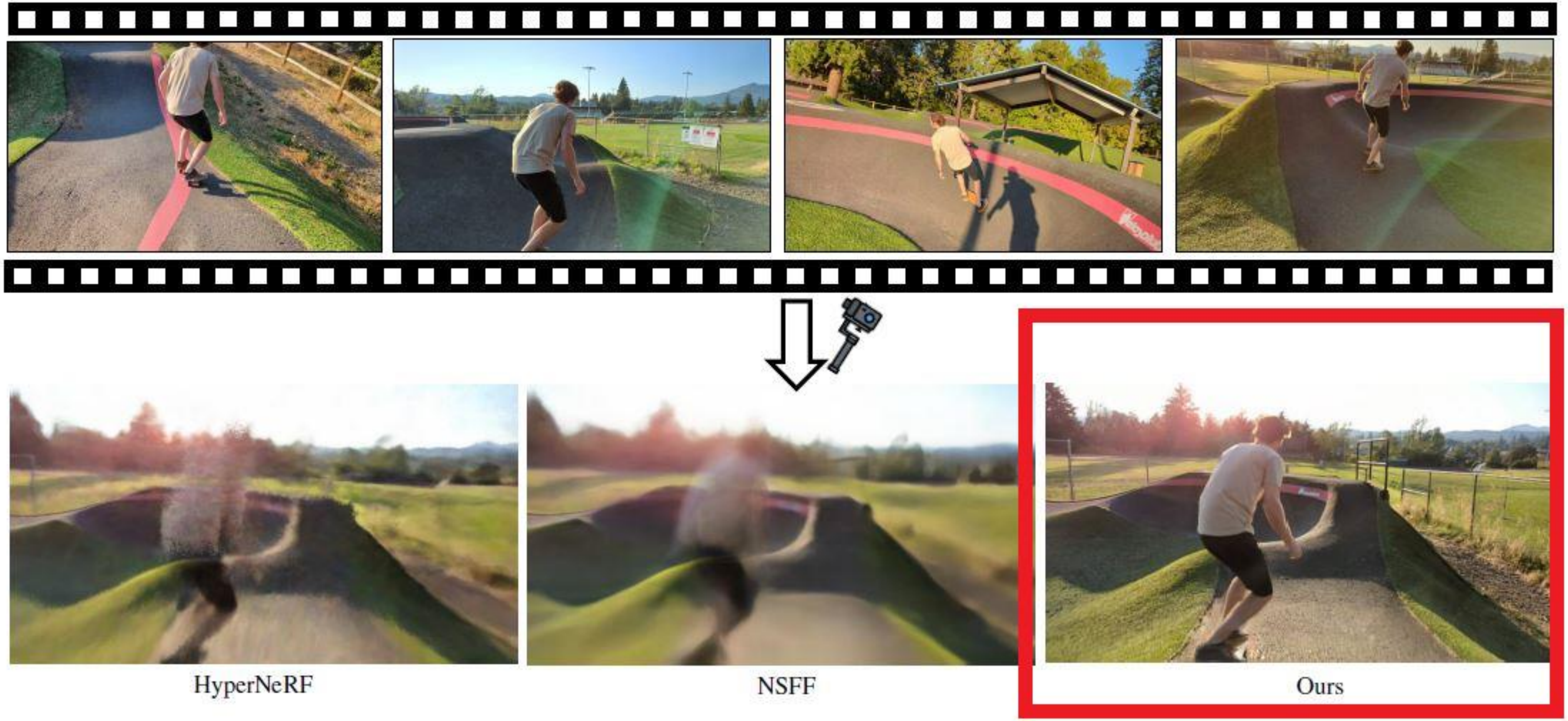
DynIBaR

Neural Dynamic Image-Based Rendering

Seungyeol Lee

Purpose of Research

- **Synthesizing Novel Views depicting a complex dynamic scene**

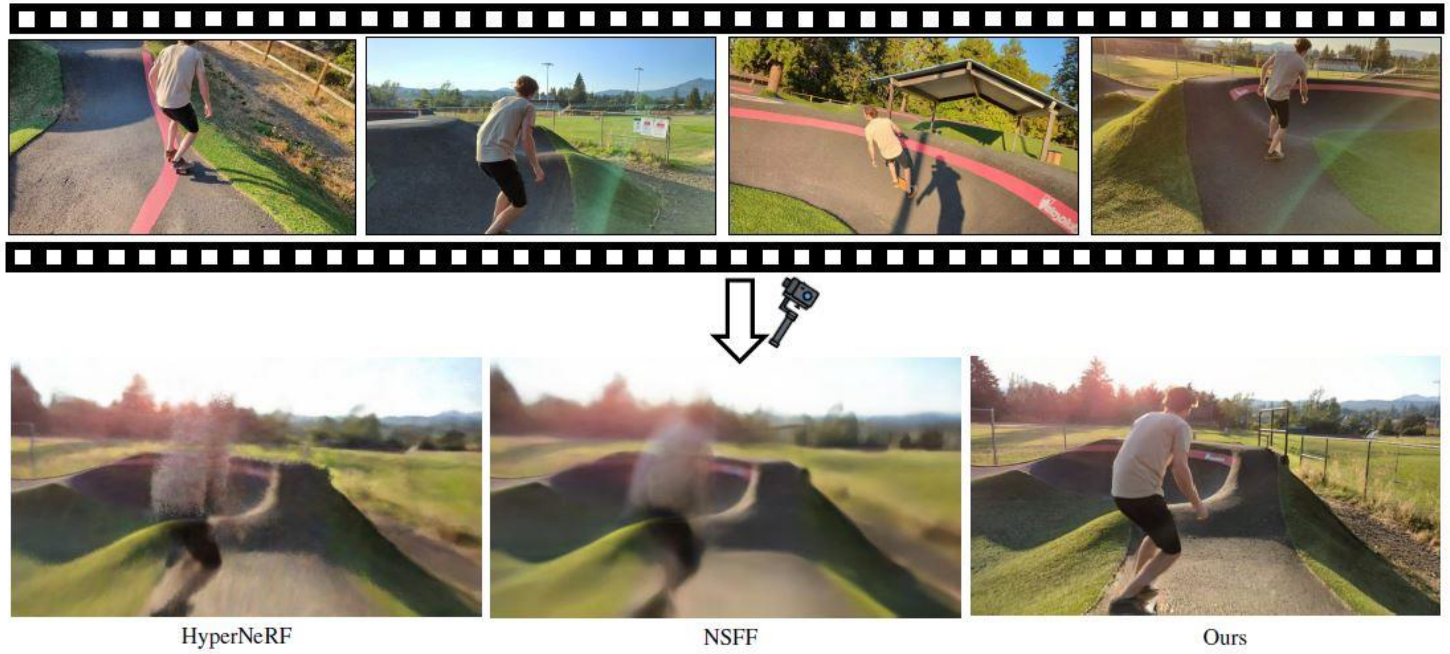


Purpose of Research

- Synthesizing Novel Views depicting a complex dynamic scene

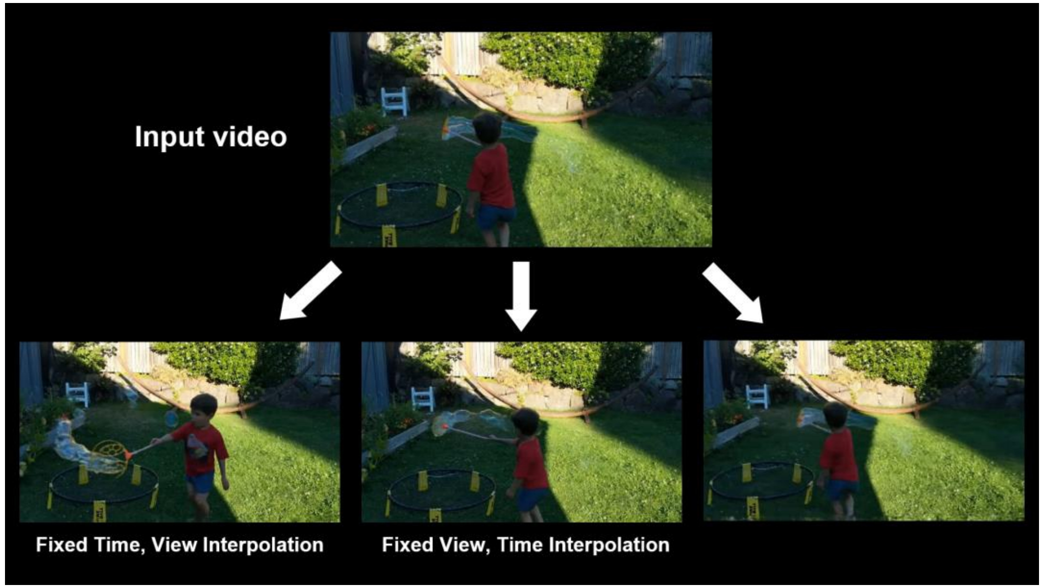
Purpose of Research

- **Novel view synthesis from a Monocular Video is Challenging**



Limitations of Previous Research

1. NSFF



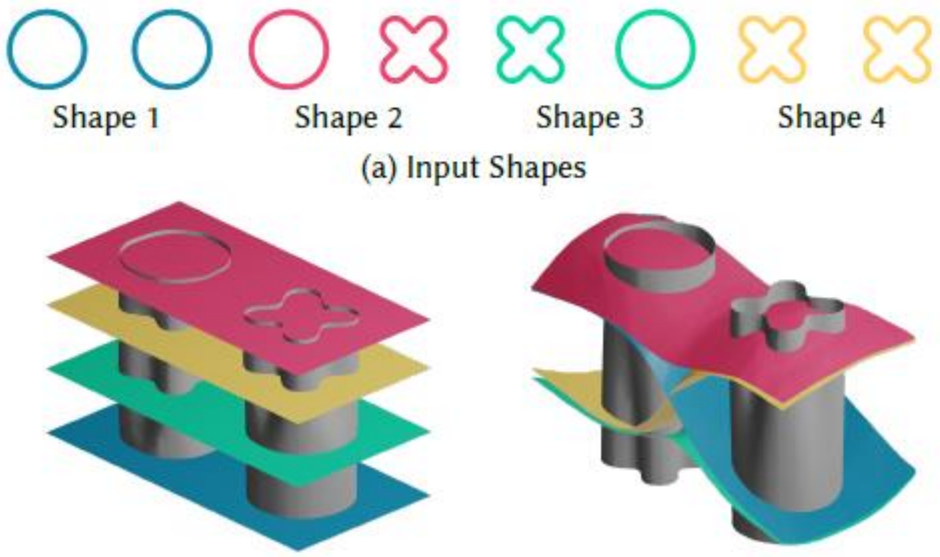
- Local scene-flow based methods

- Struggle to scale longer input videos captured with unconstrained camera motions.

- Only good performance for 1-second, forward-facing videos.

Limitations of Previous Research

2. HyperNeRF



- Construct a canonical model

- Mostly constrained to object-centric scenes with controlled camera paths

- Can fail on scenes with complex object motion.

DynIBaR

New Approach scalable to Dynamic Videos

Captured with

(1) Long time duration

(2) Unbounded scenes

(3) Uncontrolled Camera Trajectories

(4) Fast and Complex Object Motion

DynIBaR

1. Rendering Static Scenes

- Aggregate multi-view image features in “scene motion-adjusted” ray space.

- Correctly reason about spatio-temporally varying geometry and appearance.

2. Rendering Dynamic Scene Motions

- “Motion Trajectory Fields” that span multiple frames

- “Motion Trajectory Fields” represented with learned basis functions.

3. Temporal Coherence in Dynamic Scene Reconstruction

- Introduce a new temporal photometric loss

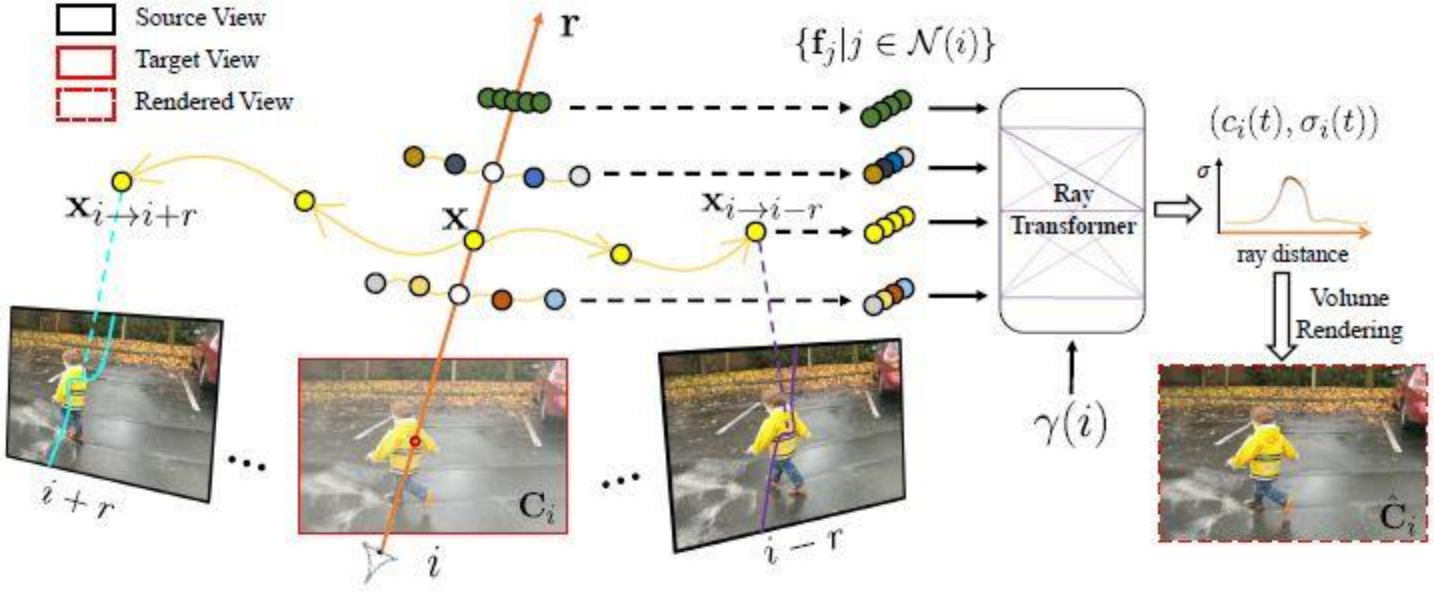
- Operated in motion-adjusted ray space

4. New IBR-based Motion Segmentation Technique

- Factor the scene into static and dynamic components

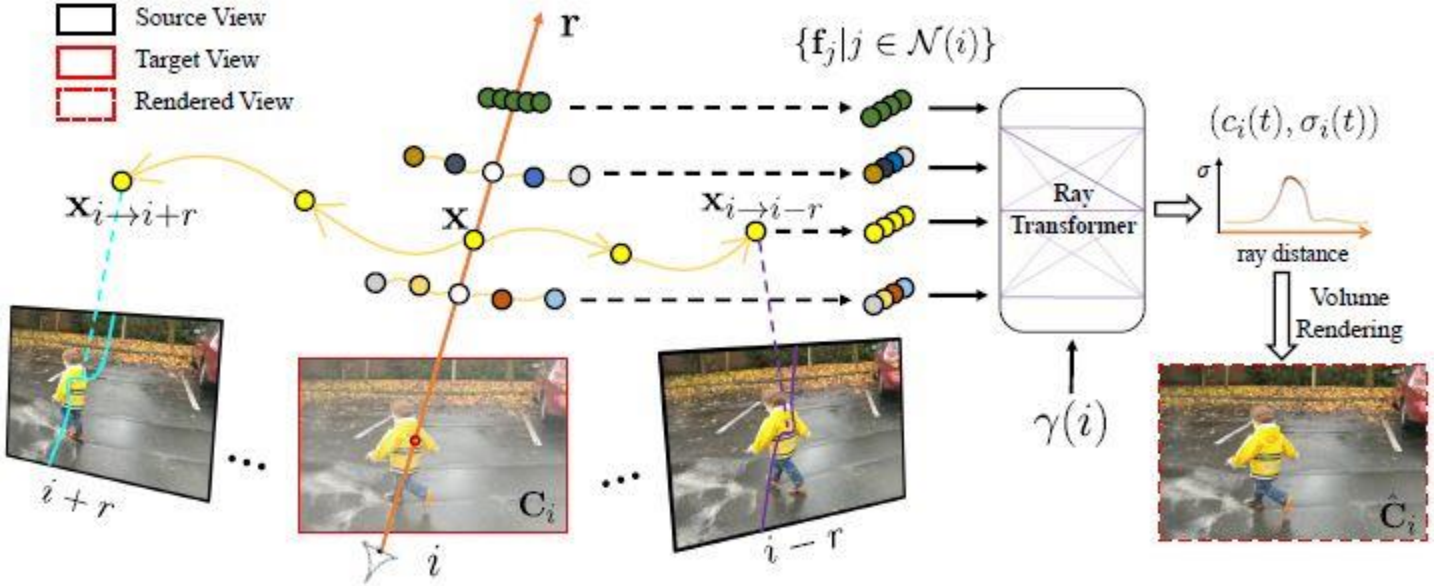
- Use Bayesian Learning Framework

Motion-adjusted Feature Aggregation



Aggregating Features extracted from temporally nearby source views

Motion-adjusted Feature Aggregation

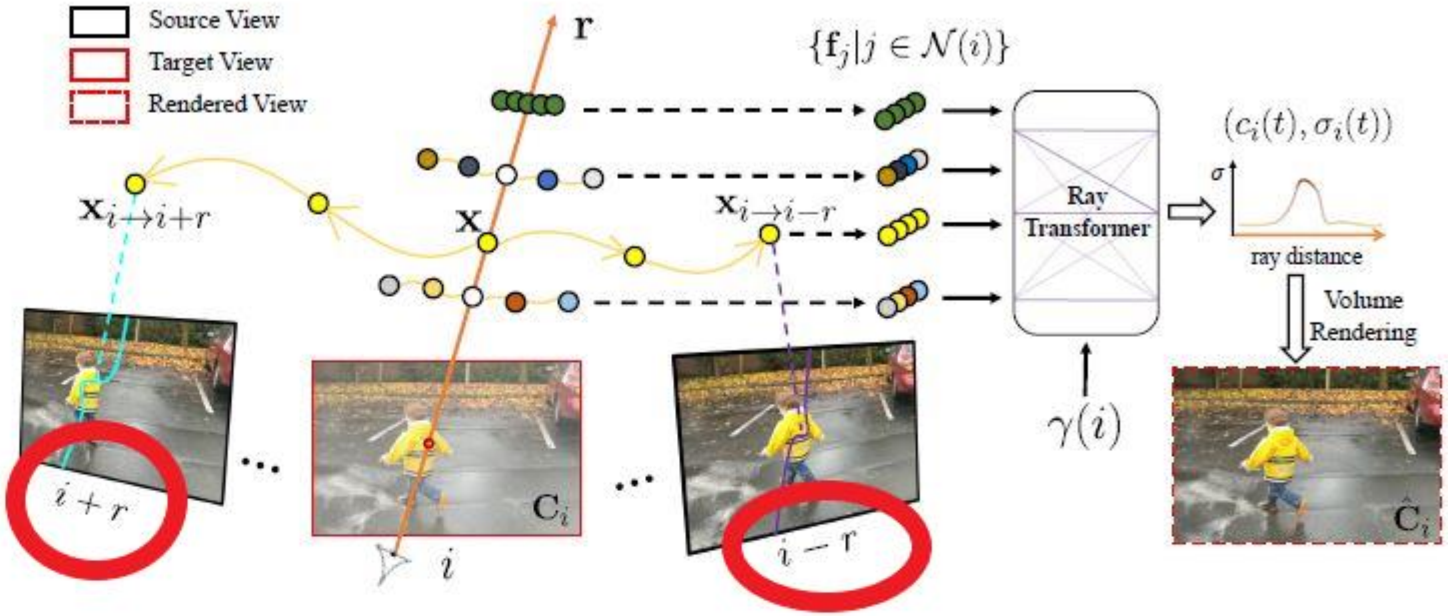


Given

$(I_1, I_2, \ldots, I_N)$ - Image Frames

$(\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N)$ - Known Camera Parameters

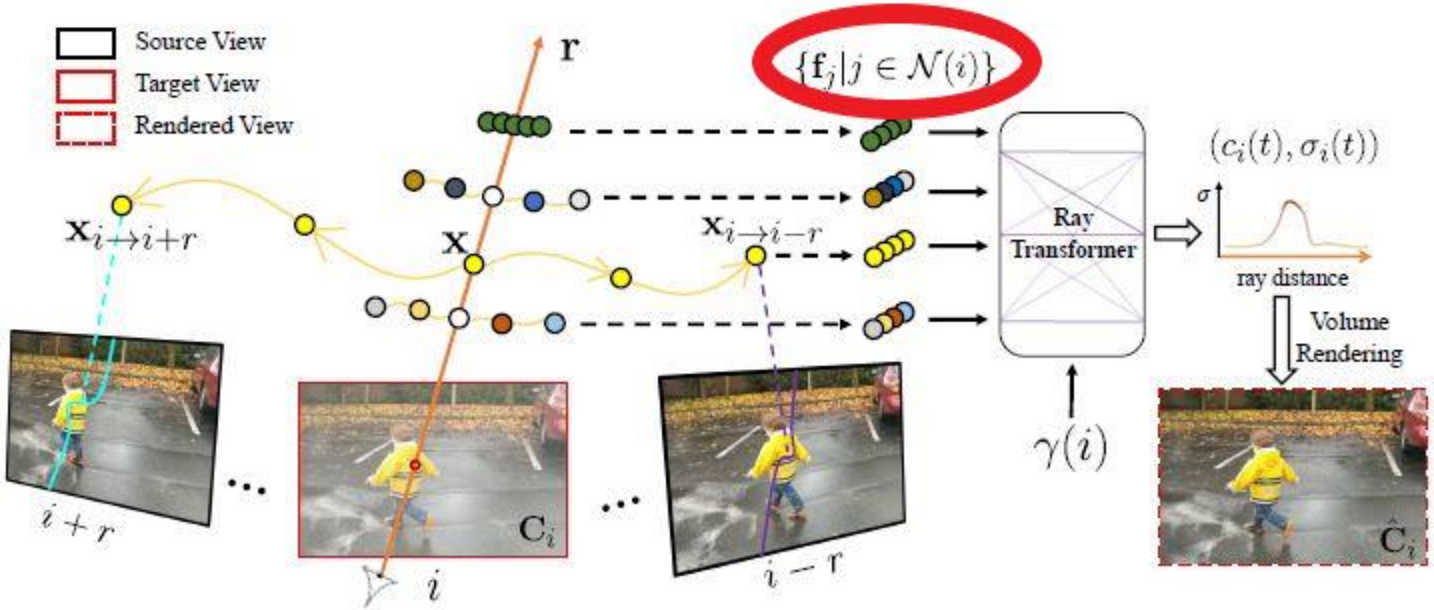
Motion-adjusted Feature Aggregation



Render an image at time i

- Identify source views I_j within a temporal radius r frames of i
- $j \in N(i) = [i-r, i+r]$

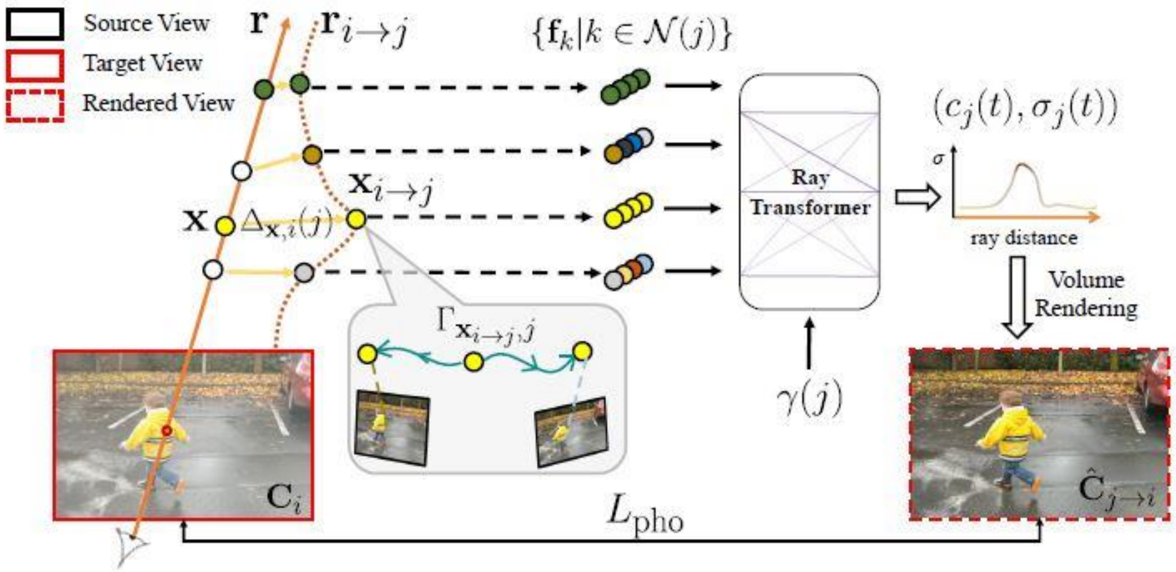
Motion-adjusted Feature Aggregation



Extract 2D Feature Map F_i

- For each source view, extract 2D feature map.
- Extracted by shared convolutional encoder network.

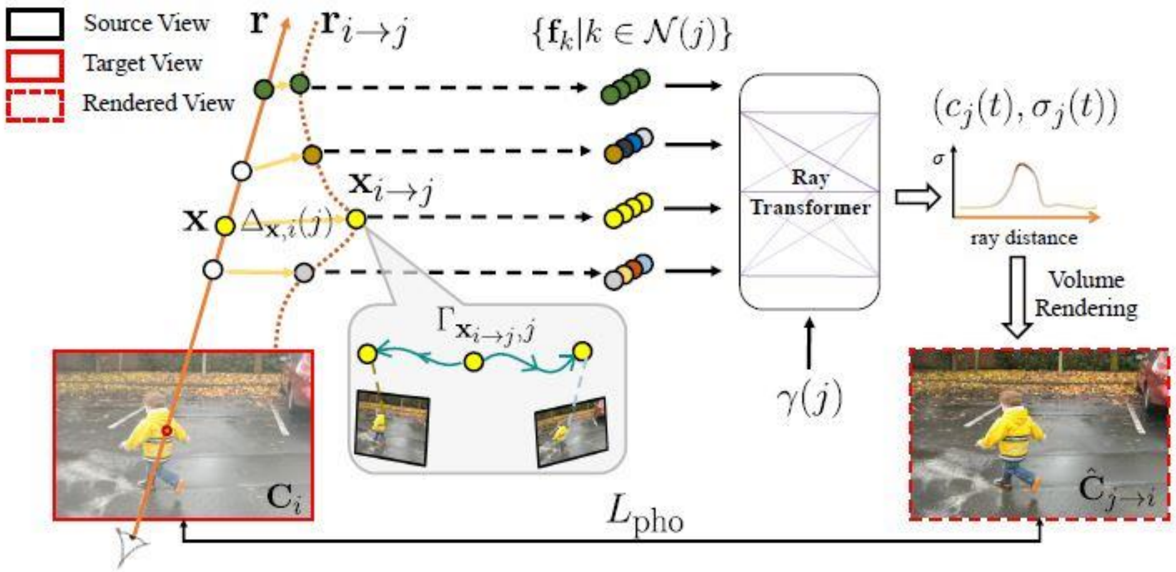
Motion-adjusted Feature Aggregation



Account for Scene Motion

- Moving scene elements lead to inconsistent feature aggregation.
- So, we perform motion-adjusted feature aggregation.

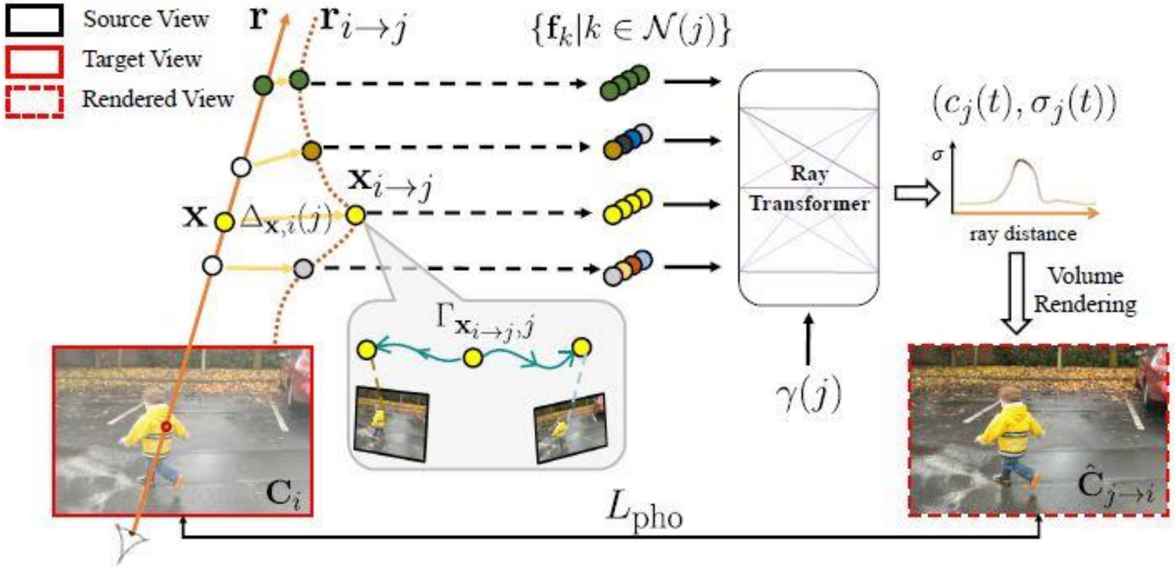
Motion Trajectory Fields



Trajectory Coefficients

$$\{\phi_i^l(\mathbf{x})\}_{l=1}^{L} = G_{\text{MT}}\left(\gamma(\mathbf{x}), \gamma(i)\right)$$

Motion Trajectory Fields

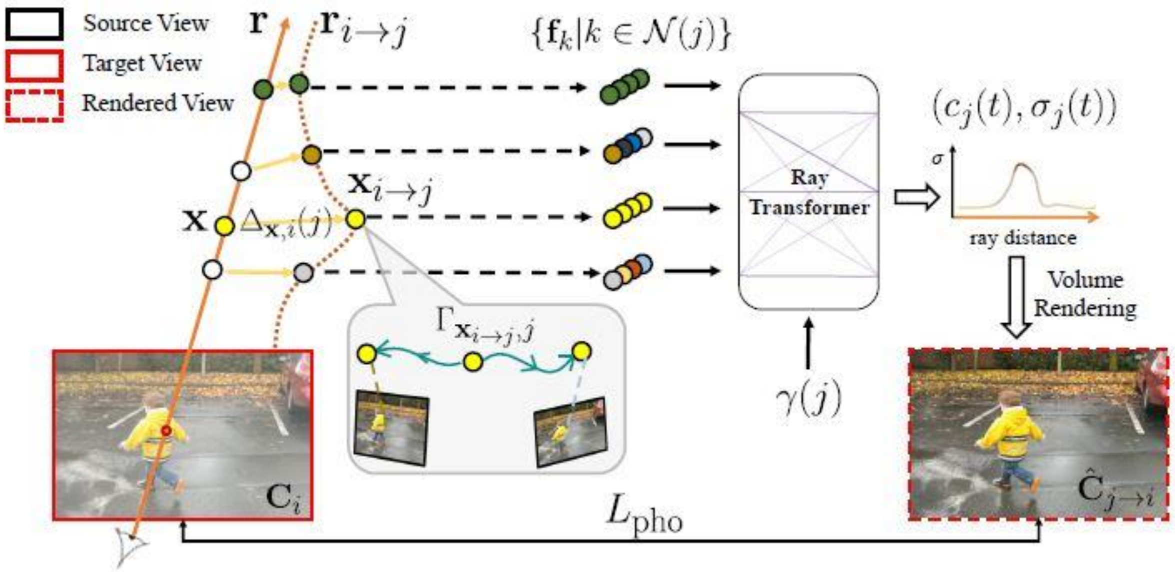


Trajectory Coefficients

$$\{\phi_i^l(\mathbf{x})\}_{l=1}^{L} = G_{\text{MT}}\left(\gamma(\mathbf{x}), \gamma(i)\right)$$

Given 3D point $\mathbf{x}$ along target ray r at time i

Motion Trajectory Fields

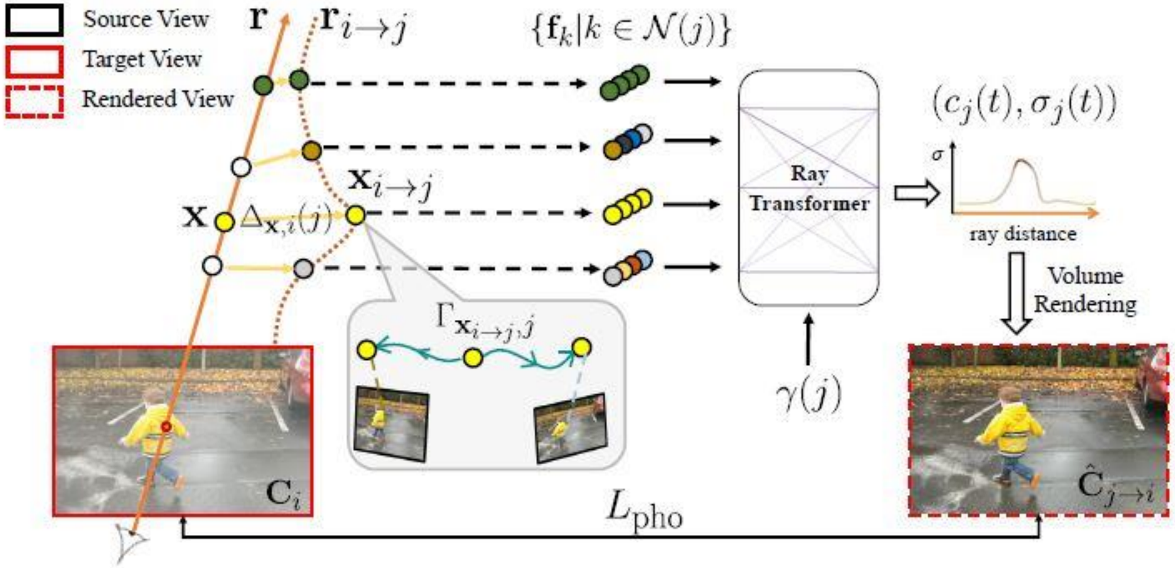


Trajectory Coefficients

$$\{\phi_i^l(\mathbf{x})\}_{l=1}^{L} = G_{\mathrm{MT}}\left(\gamma(\mathbf{x}), \gamma(i)\right)$$

γ - Positional Encoding

Motion Trajectory Fields

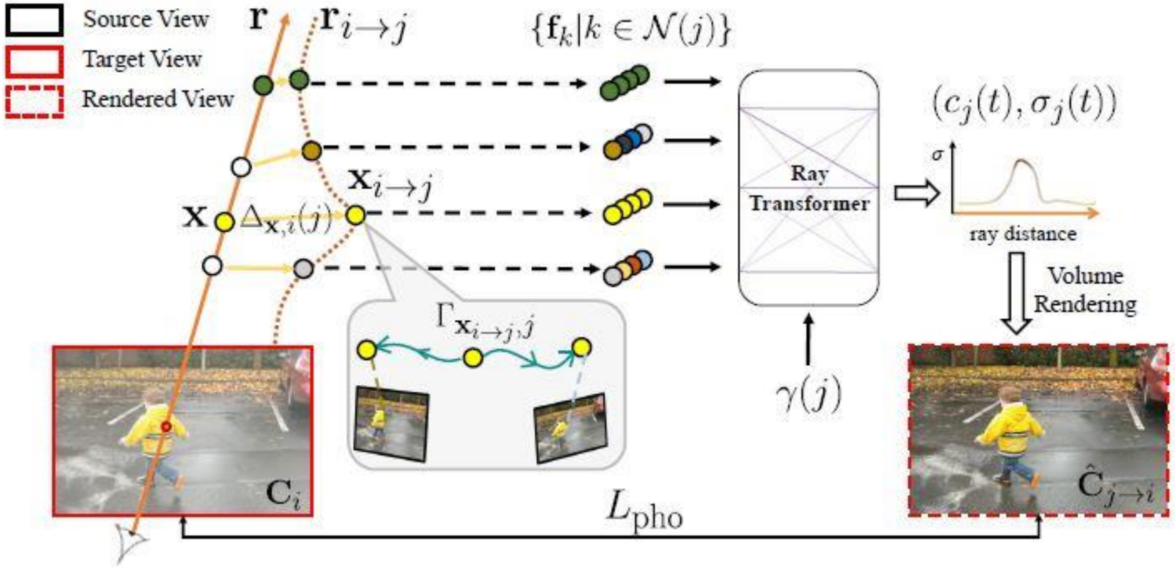


Trajectory Coefficients

$$\{\phi_i^l(\mathbf{x})\}_{l=1}^{L} = G_{\text{MT}}(\gamma(\mathbf{x}), \gamma(i))$$

G_{MT} - Motion Trajectory MLP

Motion Trajectory Fields

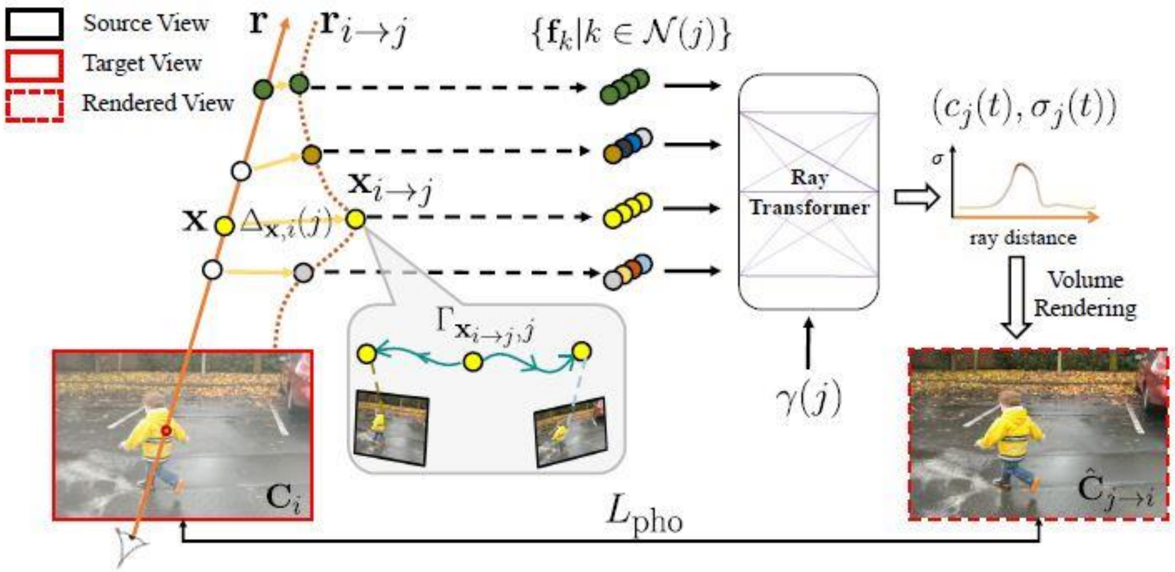


Trajectory Coefficients

$$\{\underline{\phi_i^l(\mathbf{x})}\}_{l=1}^{L} = G_{\mathrm{MT}}\left(\gamma(\mathbf{x}), \gamma(i)\right)$$

$\phi_i^l \in \mathcal{R}^3$ - Basis Coefficients for x, y, z using the motion basis

Motion Trajectory Fields

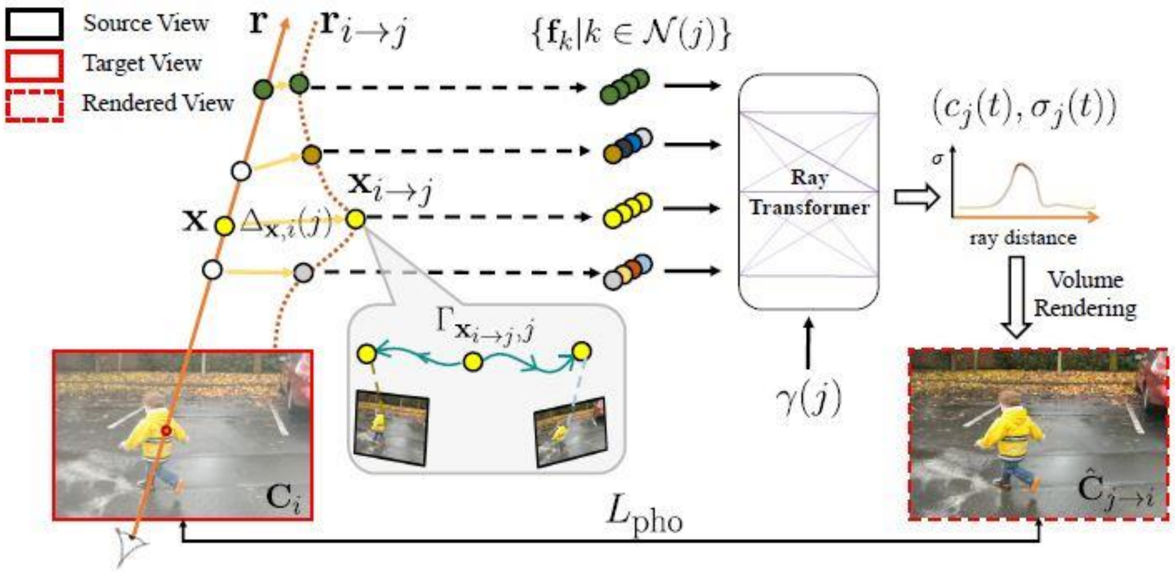


Global Learnable Motion Basis

$$\{h_i^l\}_{l=1}^{L}$$

Optimized jointly with the MLP ($h_i^l \in \mathcal{R}$)

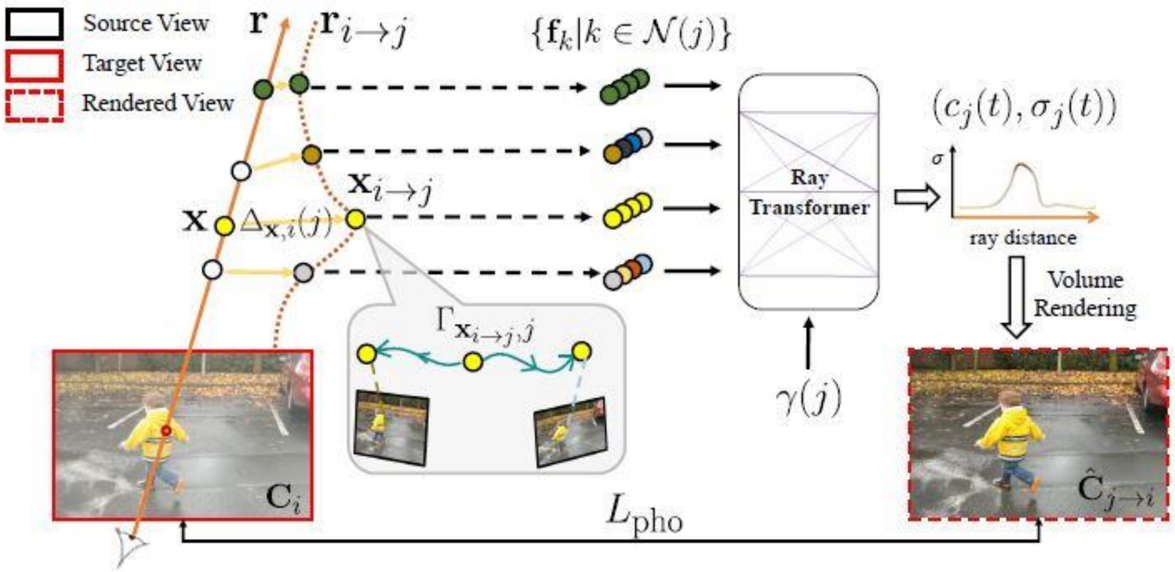
Motion Trajectory Fields



Motion Trajectory of $\mathbf{x}$

$$\Gamma_{\mathbf{x},i}(j) = \sum_{l=1}^{L} h_j^l \phi_i^l(\mathbf{x})$$

Motion Trajectory Fields

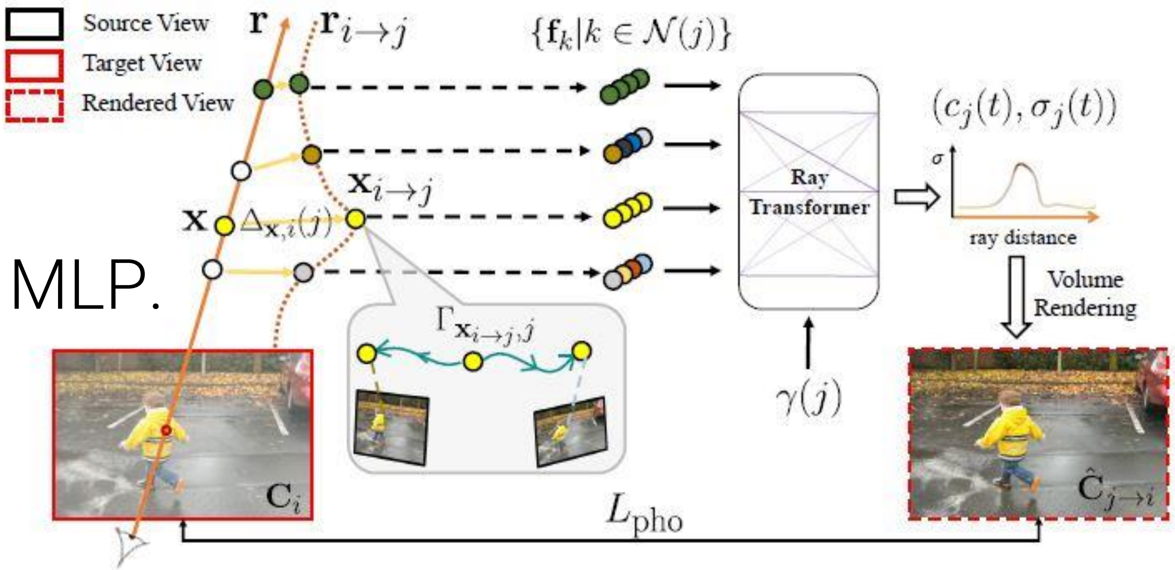


Relative displacement between $\mathbf{x}$ and $\mathbf{x}_{i\to j}$

$$\Delta_{\mathbf{x},i}(j) = \Gamma_{\mathbf{x},i}(j) - \Gamma_{\mathbf{x},i}(i).$$

Motion Trajectory Fields

Summarize



1. Source features are fed to a shared MLP.

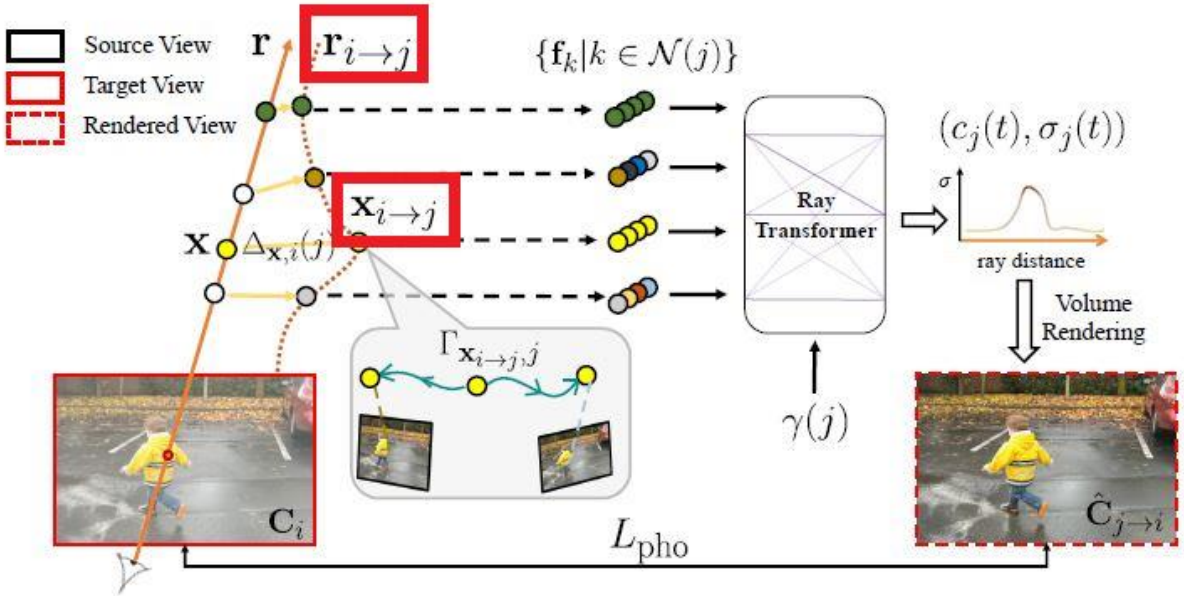
2. Shared MLP produces a single feature vector at each 3D sample point.

3. Ray Transformer processes aggregated features produced by shared MLP.

4. Ray Transformer predicts $(\mathbf{c}_i, \sigma_i)$ (per-sample colors and densities.)

5. We use NeRF volume rendering to obtain a final pixel color $\hat{\mathbf{C}}_i(\mathbf{r})$.

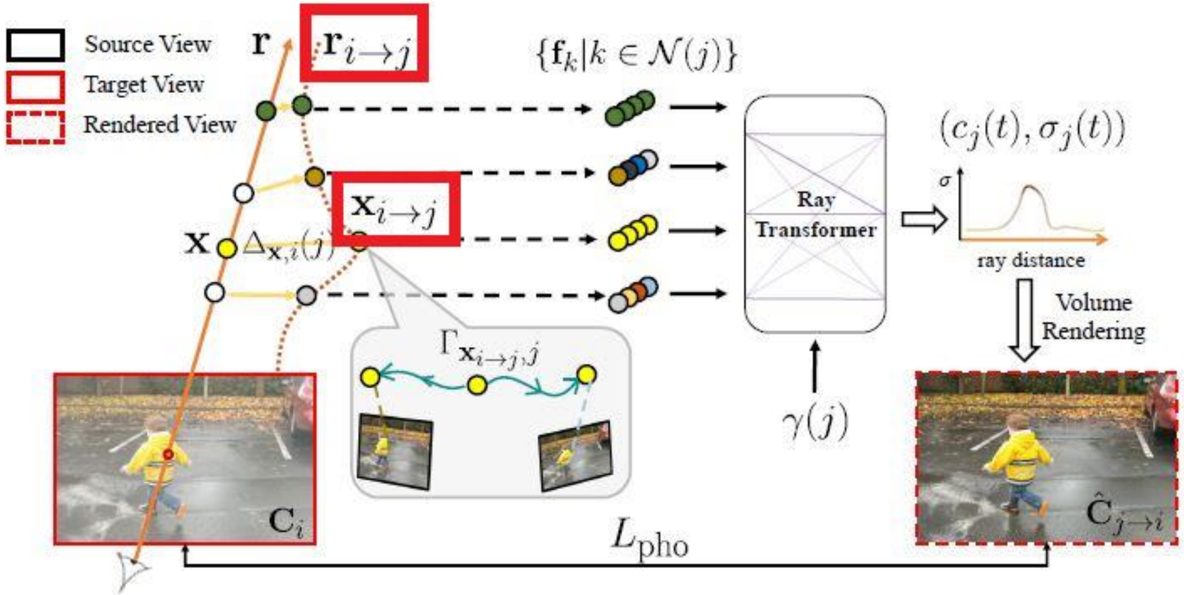
Cross-time Rendering



Points $\mathbf{x}_{i\to j}$ along motion-adjusted ray $\mathbf{r}_{i\to j}$

- Treat them as if they lie along a ray at time j

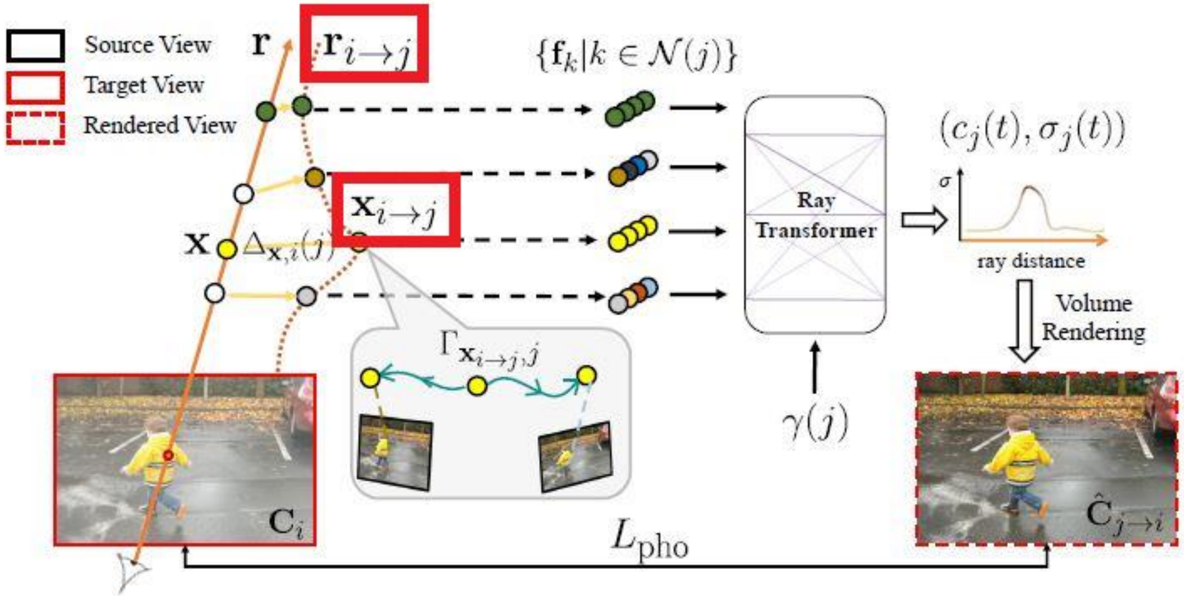
Cross-time Rendering



Motion-disocclusion-aware RGB reconstruction loss $\mathcal{L}_{\text{pho}}$

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j\to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j\to i}(\mathbf{r})).$$

Cross-time Rendering

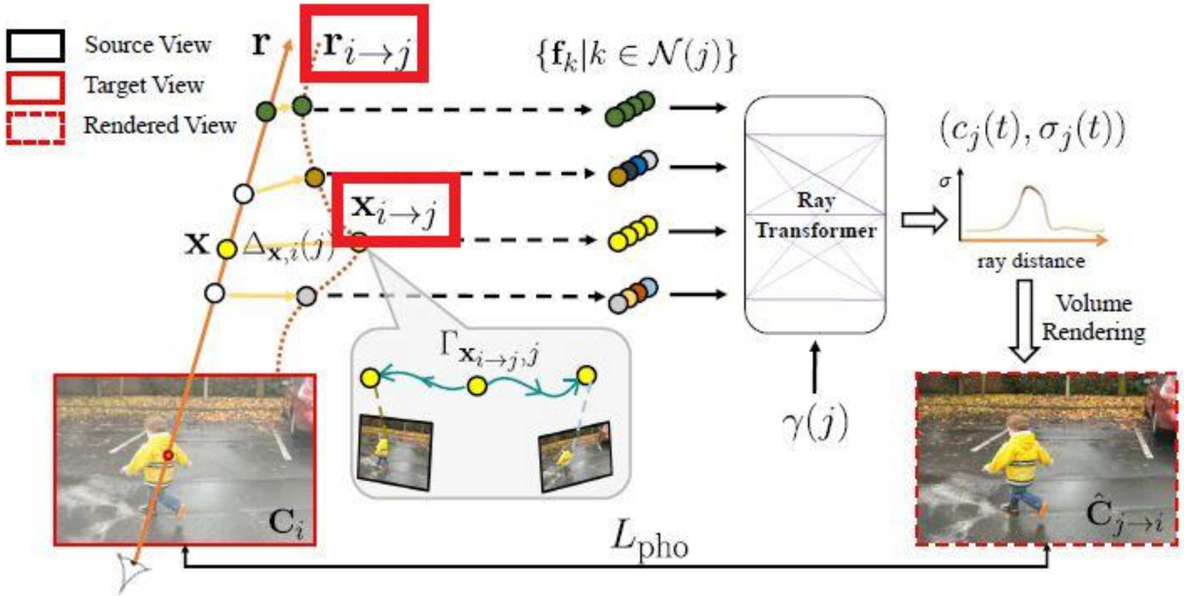


Motion-disocclusion-aware RGB reconstruction loss $\mathcal{L}_{\text{pho}}$

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j\to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j\to i}(\mathbf{r})).$$

$\hat{\mathbf{W}}_{j\to i}(\mathbf{r})$ - Motion Disocclusion Weight

Cross-time Rendering



Motion-disocclusion-aware RGB reconstruction loss $\mathcal{L}_{\text{pho}}$

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j\to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j\to i}(\mathbf{r})).$$

ρ. - Generalized Charbonnier loss for RGB

Combining Static and Dynamic Models

Dynamic Content (c_i, σ_i) with a time-varying model

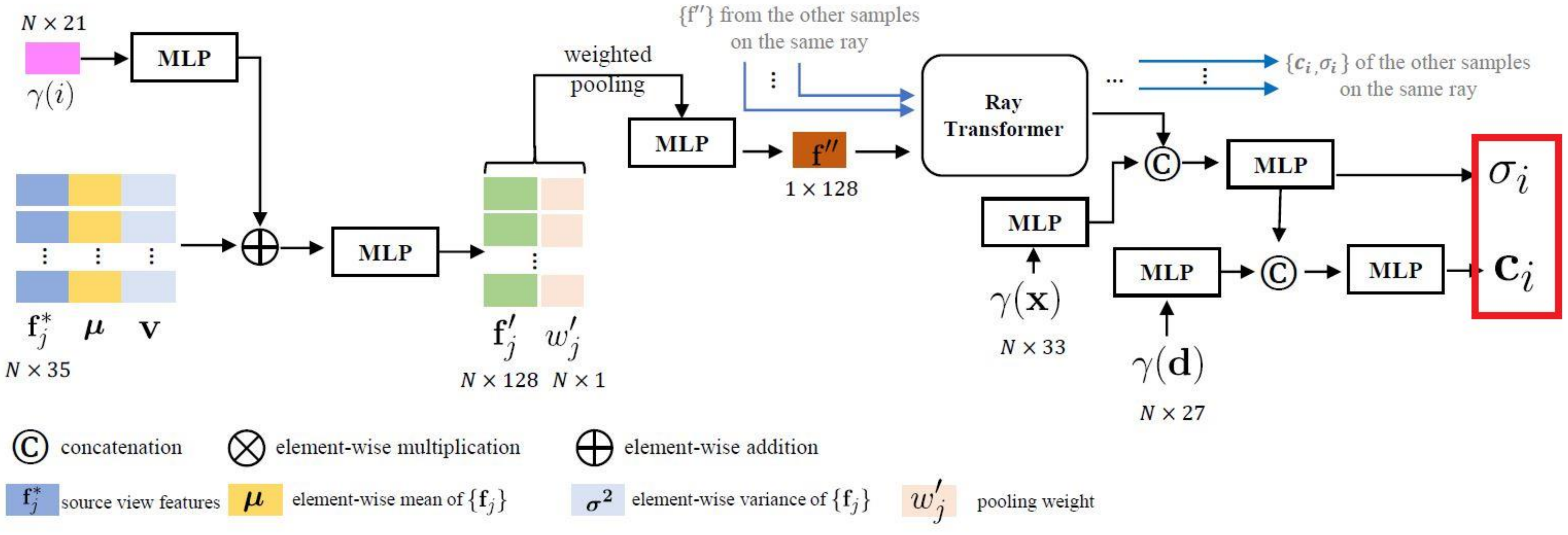


Figure 3. **Network architecture of our time-varying dynamic representation.**

Combining Static and Dynamic Models

Static content (c, σ) with a time-invariant model

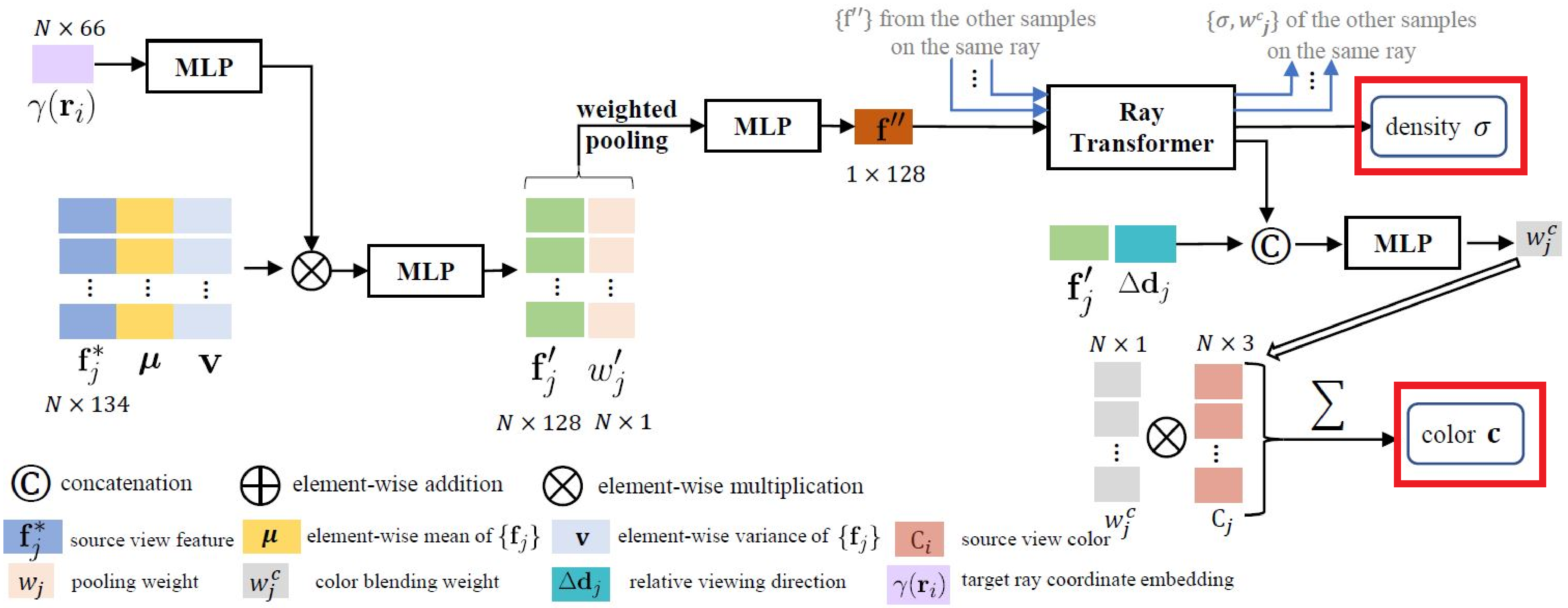


Figure 4. **Network architecture of our time-invariant static representation.**

Combining Static and Dynamic Models

Combined Dynamic and Static Predictions

$\hat{\mathbf{C}}^{\text{st}}$: color estimated by time-invariant model

$\hat{\mathbf{C}}_i^{\text{dy}}$: color estimated by time-varying model

$\hat{\mathbf{C}}_i^{\text{full}}$: color rendered by combining dynamic and static predictions

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j \to i}^{\text{full}}(\mathbf{r}))$$

Image-Based Motion Segmentation

New Motion Segmentation Module

- Segmentation masks for supervising dynamic and static scene representations

$\hat{\mathbf{B}}^{\mathrm{st}}$: pixel color rendered by IBR-Net with static scene content

$$\hat{\mathbf{B}}_i^{\mathrm{dy}}, \boldsymbol{\alpha}_i^{\mathrm{dy}}, \boldsymbol{\beta}_i^{\mathrm{dy}} = D(I_i).$$

- D : 2D-convolutional encoder-decoder network
- $\hat{\mathbf{B}}_i^{\mathrm{dy}}$: RGB image from D and input frame
- $\boldsymbol{\alpha}_i^{\mathrm{dy}}$: 2D opacity map from D and input frame
- $\boldsymbol{\beta}_i^{\mathrm{dy}}$: confidence map from D and input frame

Image-Based Motion Segmentation

Full Reconstructed Image

$$\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) = \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r})\hat{\mathbf{B}}_i^{\text{dy}}(\mathbf{r}) + (1 - \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r}))\hat{\mathbf{B}}^{\text{st}}(\mathbf{r}).$$

$\hat{\mathbf{B}}^{\text{st}}$: pixel color rendered by IBR-Net with static scene content

$\hat{\mathbf{B}}_i^{\text{dy}}$: RGB image from D and input frame

$\boldsymbol{\alpha}_i^{\text{dy}}$: 2D opacity map from D and input frame

β_i^{dy} : confidence map from D and input frame

Image-Based Motion Segmentation

Segmentation Loss

$$\mathcal{L}_{\text{seg}} = \sum_{\mathbf{r}} \log \left(\boldsymbol{\beta}_i^{\text{dy}}(\mathbf{r}) + \frac{||\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) - \mathbf{C}_i(\mathbf{r})||^2}{\boldsymbol{\beta}_i^{\text{dy}}(\mathbf{r})} \right)$$

- Observations with a Cauchy distribution with $\boldsymbol{\beta}_i^{\text{dy}}$

- Weighted loss taking the negative log-likelihood of the observations

Image-Based Motion Segmentation

Segmentation Mask Loss

$$\mathcal{L}_{\text{mask}} = \sum_{\mathbf{r}} (1 - M_i)(\mathbf{r})\rho(\hat{\mathbf{C}}^{\text{st}}(\mathbf{r}), \mathbf{C}_i(\mathbf{r})) + \sum_{\mathbf{r}} M_i(\mathbf{r})\rho(\hat{\mathbf{C}}_i^{\text{dy}}(\mathbf{r}), \mathbf{C}_i(\mathbf{r}))$$

- M_i : Masks with time-varying and time-invariant models

- Perform to obtain masks to turn off the loss near mask boundaries

Regularization

Regularization scheme

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{MT}} + \mathcal{L}_{\text{cpt}}$$

$\mathcal{L}_{\text{data}}$: Data-Driven loss consisting of l_1 monocular depth and optical flow consistency

$\mathcal{L}_{\text{MT}}$: Motion-trajectory regularization to be cycle-consistent and spatio-temporally smooth

$\mathcal{L}_{\text{cpt}}$: Compactness prior that encourages the scene decomposition to be binary

Final Combined Loss

$$\mathcal{L} = \mathcal{L}_{\text{pho}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{reg}}$$

Experiments

Evaluation Metrics

- PSNR, SSIM, and LPIPS

- Errors over the entire scene (Full)

- Errors restricted to moving regions (Dynamic Only)

Experiments

Dynamic Scenes Dataset

- NVIDIA dataset

- UCSD dataset

- Qualitative Comparison with DVS, HyperNeRF, and NSFF

Experiments

Qualitative Comparisons on Dynamic Scenes Dataset

NVIDIA dataset

UCSD dataset

Experiments

Quantitative Comparisons on Dynamic Scenes Dataset

Methods	Full			Dynamic Only		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
Nerfies [49]	0.823	24.32	0.096	0.595	18.45	0.234
HyperNeRF [50]	0.859	25.10	0.095	0.618	19.26	0.212
DVS [19]	0.943	30.64	0.075	0.866	26.57	0.096
NSFF [35]	0.952	31.75	0.034	0.851	25.83	0.115
Ours	**0.983**	**36.47**	**0.014**	**0.909**	**28.01**	**0.042**

Experiments

In-the-wild videos

- Straightforward modification to NeRF’s Blender Dataset

- Designed to probe aliasing and scale-space reasoning

- Qualitative Comparisons with DVS, HyperNeRF, and NSFF

Experiments

Qualitative Comparisons on In-the-wild videos

Input DVS HyperNeRF NSFF Ours

Limitations

- Relatively small viewpoint changes

- Not able to handle small fast moving objects
(due to incorrect initial depth and optical flow estimates)

- Not strictly multi-view consistent

- Sensitive to degenerate motion patterns from in-the-wild videos
(degenerate motions : object and camera motion is colinear)