A. Additional Details

**Shape Regularization.** We apply eikonal regularization to force the norm of the first order derivative of signed distances \( d \) to be close to 1,

\[
\mathcal{L}_{\text{eikonal}} = (\| \nabla \text{MLP}_{\text{SDF}}(X) \|- 1)^2. \tag{1}
\]

Eikonal loss forces the reconstruction to be a valid surface and empirically improves the surface reconstruction quality.

**Pose, Deformation, and Appearance Smoothness.** We would like the time-varying articulated pose, deformation, and appearance codes \( \{\theta, \omega_d, \omega_a\} \) to vary smoothly within a video. To accomplish this, we make use of time-dependent positional embeddings (similar to [11]):

\[
\omega^b = \mathbf{A}_i \mathcal{F}(t) \tag{2}
\]

where \( \mathcal{F}(\cdot) \) is a 1D basis of sines and cosines with linearly-increasing frequencies at log-scale [6], and we learn separate weight matrices \( \mathbf{A}_i \in \{1, \ldots, M\} \) for each video.

B. Category Outside DensePose

We test RAC in a scenario where there is no predefined DensePose features and skeleton.

![Vehicle Category Reconstruction](image)

**Vehicle Dataset.** We employ images from multiple 4K cameras [4] that overlook urban public spaces to analyze the flow of traffic vehicles. The data are captured for 3-second bursts every few minutes, and only images with notable changes are stored. We extracted 365 car videos from the dataset to build the category model. The dataset contains wide variation in vehicle categories like pickup trucks, construction vehicles etc on which traditional model based approached perform poorly.

**Camera Pose Initialization.** As there is no DensePose model for cars, we took a two-stage approach to first coarsely register a few car videos with manual viewpoint annotation and then train a single-image viewpoint network to predict viewpoints for the rest of the videos. The camera viewpoints are roughly annotated for each frame (with around 30 degree rotation error). Annotation for a 100 frame video takes around 30 seconds. We found annotating 20 cars to be sufficient to train a viewpoint estimator that generalizes to other cars.

**Results.** We show the reconstruction results of car videos in Fig. 1. Please visit the website for more results.

C. Evaluation on Pablo Sequence

We compare with baselines on the Pablo sequence, which is part of the public MonoPerfCap [10] dataset. Our method optimizes the Pablo sequence together with the rest of our 47 human videos. After differentiable rendering optimization, we extract meshes for the Pablo sequence and compare with the 3D ground-truth for evaluation.

**Metrics.** We follow the evaluation protocol of MonoClothCap [9] and compute the average point-to-surface distances in the clothing region. The clothing region (the shirt and shorts) is obtained by manual segmentation on the ground-truth surface mesh.

**Results.** We show quantitative comparisons in Tab. 1 and refer the reader to the qualitative results in Fig. 8 of the main draft. Our method outperforms PiFuHD [5], Tex2Shape (T2S) [1], both of which are single-view human shape pre-

<table>
<thead>
<tr>
<th>Method</th>
<th>MPCap*</th>
<th>MCCap</th>
<th>PiFuHD</th>
<th>T2S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chamfer</td>
<td>14.6</td>
<td>17.9</td>
<td>26.5</td>
<td>27.7</td>
</tr>
</tbody>
</table>
dictors trained on 3D scans of humans. Our method does not use 3D data to train but performs test-time optimization on 47 human videos. Our method is slightly worse than MonoClothCap (MCCap) [9] that uses a parametric human body model (SMPL), and worse than MonoPerfCap (MP-Cap), which uses a prescanned template. Both parametric body model and personalized shape template provides a strong shape prior, while our method does not rely on any shape prior.

D. Difference from prior works

We highlight the difference from previous work in Tab. 2. In terms of shape modeling, HyperNeRF [3] and HumanNeRF [7] reconstruct a single scene or instance, while learns a space of category shapes. For skeleton modeling, CASA [8] is optimized per-instance, while learns a shared space over a category of skeletons (with different bone lengths). For background modeling, NeRF++ [12] assumes a static scene and does not use background to help object segmentation and reconstruction. NeFace [2] treats background as a static image, while we represent the background as a NeRF, which generalizes to videos captured by a moving camera.

<table>
<thead>
<tr>
<th>Method</th>
<th>Shape</th>
<th>Motion</th>
<th>Background</th>
<th>3D Data/Pose</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeRF++</td>
<td>N.A.</td>
<td>N.A.</td>
<td>NeRF</td>
<td>No</td>
</tr>
<tr>
<td>NeFace</td>
<td>Instance</td>
<td>Conditional</td>
<td>Image</td>
<td>No</td>
</tr>
<tr>
<td>HyperNeRF</td>
<td>Instance</td>
<td>Fields+Conditional</td>
<td>N.A.</td>
<td>No</td>
</tr>
<tr>
<td>BANMo</td>
<td>Instance</td>
<td>Control Points</td>
<td>N.A.</td>
<td>No</td>
</tr>
<tr>
<td>CASA</td>
<td>Instance</td>
<td>Instance Skeleton</td>
<td>N.A.</td>
<td>Yes</td>
</tr>
<tr>
<td>HumanNeRF</td>
<td>Instance</td>
<td>Instance Skeleton</td>
<td>N.A.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Category</td>
<td>Category Skeleton</td>
<td>NeRF</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2. Difference between prior works and .

References


