Distilling Neural Fields for Real-Time Articulated Shape Reconstruction

Jeff Tan Gengshan Yang Deva Ramanan
Carnegie Mellon University

Figure 1. By distilling knowledge from dynamic NeRFs fitted to offline video data at scale [16, 44], we present a method to train category-specific real-time video shape predictors, which output *temporally-consistent* viewpoint, articulation, and appearance given casual input videos. Our method replaces expensive test-time optimization with a single forward pass, allowing real-time inference on a RTX-3090 GPU. Compared to existing model-based methods for reconstructing humans and animals in motion [13, 18, 31], our method does not require pre-defined 3D templates or ground-truth 3D data to train. (Project page: https://jefftan969.github.io/dasr).

Abstract

We present a method for reconstructing articulated 3D models from videos in real-time, without test-time optimization or manual 3D supervision at training time. Prior work often relies on pre-built deformable models (e.g. SMAL/SMPL), or slow per-scene optimization through differentiable rendering (e.g. dynamic NeRFs). Such methods fail to support arbitrary object categories, or are unsuitable for real-time applications. To address the challenge of collecting large-scale 3D training data for arbitrary deformable object categories, our key insight is to use off-the-shelf video-based dynamic NeRFs as 3D supervision to train a fast feed-forward network, turning 3D shape and motion prediction into a supervised distillation task. Our temporal-aware network uses articulated bones and blend skinning to represent arbitrary deformations, and is self-supervised on video datasets without requiring 3D shapes or viewpoints as input. Through distillation, our network learns to 3D-reconstruct unseen articulated objects at interactive frame rates. Our method yields higher-fidelity 3D reconstructions than prior real-time methods for animals, with the ability to render realistic images at novel viewpoints and poses.

1. Introduction

We are interested in building high-quality animatable models of articulated 3D objects from videos in real time. One promising application is virtual and augmented reality, where the goal is to create high-fidelity 3D experiences from images and videos captured live by users. For rigid scenes, structure from motion (SfM) and neural rendering can be used to build accurate 3D cities and landmarks from Internet image collections [1, 20, 33]. For articulated objects such as friends and pets, many works parameterize the range of motions using category-specific templates such as SMPL [18] for humans and SMAL [4] for quadruped animals. Although these methods can be trained on large-scale video datasets, they rely on parametric body template models built from extensive real-world 3D scans: these body models are not easy to generate for diverse categories in the wild such as clothed humans or pets with distinct morphologies, which are often the focus of user content.

Inspired by the breakthrough success of neural radiance fields [21], many works reconstruct arbitrary articulated objects in an analysis-by-synthesis framework [16, 27, 28, 30, 36, 44] by defining time-dependent 3D warping fields and establishing long-range correspondences on top of canonical shape and appearance models. These methods output high-quality reconstructions of arbitrary objects without 3D data or pre-defined templates, but the output representations are scene-specific and often require *hours* to compute from scratch on unseen videos - an unacceptable cost for real-time VR/AR tasks. We are thus interested in dynamic 3D reconstruction algorithms that achieve the best of both worlds: the speed of template-based models and the quality and generalization ability of dynamic NeRFs.

To achieve this, our key insight is remarkably simple: *we train category-specific feed-forward 3D predictors at scale by self-supervising them with dynamic NeRF "teachers" fitted to offline video data.*

By leveraging scene-fitted dynamic NeRFs for 3D supervision at scale, our method learns a feed-forward predictor for appearance, 3D shape, and articulations of non-rigid objects from videos. Our learned 3D models use linear blend skinning to express articulations, allowing it to be animated by manipulating bone transformations. We address three key challenges in our work: (1) how to supervise feed-forward models with internal representations of dynamic NeRFs, (2) how to produce temporally consistent predictions of pose, articulation, and appearance, and (3) how to build efficient systems for real-time reconstruction.

2. Related Work

Template-Based Dynamic Reconstruction A large body of work uses parametric body models [18, 48, 49] to recover 3D shapes and motions for human and animal reconstruction, given a single image as input [2, 3, 13, 32]. These models are built from registered 3D scans of real humans or toy animals, and achieve great success in reconstructing categories for which large volumes of ground-truth 3D data are available (especially human reconstruction). However, it is challenging to apply these methods to arbitrary categories with diverse morphologies, especially where 3D data is limited. Our work aims to generalize these approaches to arbitrary articulated object categories without requiring ground-truth 3D data or pre-registered 3D scans during training.

Template-Free Dynamic Reconstruction Several methods build deformable 3D models without templates by recovering shapes and poses from internet-scale 2D image collections, using weak supervision such as keypoints and object silhouettes from off-the-shelf models or human annotators [6, 10, 14, 34]. As it is inherently ambiguous to reconstruct 3D outputs from the sparse and limited 2D observations available in images, these methods must leverage strong data priors and apply heavily regularization to ensure reasonable outputs, often resulting in blurry or over-smoothed shapes and textures. Leveraging the temporal context available in videos can help these methods learn temporally consistent results [38], however the output quality is still low perhaps due to over-regularization.

Neural Radiance Fields Neural fields are a powerful method for 3D reconstruction using 2D image supervision, achieving state-of-the-art quality on both static and dynamic scenes. Although historically limited to rigid scenes with known cameras [20, 21, 36], recent works extend NeRF to dynamic scenes by deforming view-space points to a canonical space over time, using time-dependent 3D warping fields and dense correspondences [5, 16, 17, 27, 30]. Dynamic NeRFs are able to learn high-fidelity animatable 3D models from several casual videos capturing the same object instance [42–44], by leveraging the temporal context available in videos. Unfortunately, dynamic NeRFs are slow and require optimization from scratch on unseen videos at test-time. Although architectures exist to speed up static NeRFs using explicit voxel grids [46] or hash-table caching [22], more work is required to speed up dynamic NeRFs by similar factors due to the additional overheads of time-dependent warping and correspondence matching. Our aim is to leverage the high-quality outputs of dynamic NeRFs to supervise a fast and lightweight architecture for articulated 3D reconstruction.

3. Method

In order to train category-specific feed-forward 3D predictors from dynamic NeRF teacher models, we combine a single-frame image encoder that regresses viewpoint, shape, and appearance from images, and a temporal encoder that reasons about these predictions over time. In this section, we describe the problem setup, scene representation, network architecture, training procedure, and losses.

3.1. Problem Setup

Given an input video centered on an articulated object, we train a feed-forward network to predict the viewpoint, articulations, and appearance, which are used to render a posed and textured object model. Our network is supervised on the pseudo-ground-truth outputs of a dynamic NeRF teacher, which builds animatable 3D models from casually collected videos including shape, appearance, and time-varying articulations. Our particular implementation uses BANMo [44] as the teacher, as it has public code and yields good results on humans and quadrupeds. Similar to the teacher, our method requires no pre-defined shape templates, registered cameras, or 3D ground truths.

Fig. 2 summarizes our approach and how it differs from the teacher model. Sec. 3.2 and Sec. 3.3 introduce our object and motion representation, while Sec. 3.4 discusses the underlying feed-forward neural architecture. Finally, Sec. 3.5 describes the training losses used to supervise our models. In contrast to our teacher [44] which uses volume rendering (which can be slow), we render textured meshes to enable efficient rasterization.

3.2. Object Representation

Category-level shape. We model articulated objects as a canonical rest shape that is transformed by time-dependent poses and articulations. The rest shape is a category-specific triangular mesh $M = (V, F)$ that represents the mean shape of instances in the category: the faces F define the vertex connectivity and we assume it remains fixed. To initialize the rest mesh, we run marching cubes on a 128^3 grid to find the zero level set of the neural field that serves as the teacher

NeRF's implicit rest shape. Although explicit mesh representations are less expressive than implicit neural fields, we find that a feed-forward predictor benefits from the speed of rasterization and the simplicity of regressing vertex colors.

Per-instance pose and articulation. At each frame t, we model the object's viewpoint and articulation similar to the dynamic NeRF literature. The viewpoint is a root body transformation $\mathbf{G}^t \in SE(3)$, and we use neural blend skinning [9, 39] to express object articulations. For each frame, the bone configurations for neural blend skinning are parameterized by a joint angle vector $\mathbf{A}^t \in SO(3)^b$, which is predicted by our feed-forward network from input video frames. See Sec. 3.3 for more details.

Appearance. As our mesh topology is fixed, we can simply model the object's appearance as an array of per-vertex colors $\mathbf{C}^t \in \mathbb{R}^{|V|\times 3}$ at each frame t. Following the standard rasterization pipeline, barycentric coordinates are used to interpolate the vertex colors per triangle during rendering.

Rendering. We assume a weak-perspective camera projection defined by a fixed camera at the origin pointed along the negative-z axis. To render the object in video frame t, we apply the predicted viewpoint $\mathbf{G}^t \in SE(3)$ at time t, followed by the blend skinning deformation specified by predicted bone configuration $\mathbf{A}^t \in SO(3)^b$ using forward kinematics and dual-quaternion blend skinning.

3.3. Time-Varying Articulation via Blend Skinning

To represent articulated body motion, we use a neural blend skinning model to define a 3D warping field $\mathcal{W}^t$ on top of a category-level kinematic skeleton. After computing the skeleton's forward kinematics, each point is deformed by a weighted combination of per-bone transformations.

Category-level skeleton. Unlike color and 3D shape which are directly observable from imagery, an object's bone structure is much harder to infer. Automatic skeletal rigging methods [15, 26] rely heavily on shape priors, or are sensitive to input data. Fortunately, bone structures are largely fixed within categories up to slight variations in bone length and body part scale. Thus, we can use readily available generic skeletons of humans, quadrupeds, and other categories to specify the bone structure of each category. Skeletons are defined by a tree structure with $B+1$ variable-length bones and B ball joints, where $B = 19$ for humans and $B = 26$ for quadrupeds.

Forward kinematics. Each bone b has a link transformation $\mathbf{L}_b \in SE(3)$ specifying the bone length and a joint transformation $\mathbf{J}_b^t \in SE(3)$ specified by the joint angle $\mathbf{A}_b^t \in SO(3)$. The result for each link b is a sequence of alternating transformations from the skeleton's base to the link's end, where $b_1, b_2, \ldots, b$ are the parent links up to b:

$$\mathbf{T}_b^t = \mathbf{J}_b^t \mathbf{L}_b \ldots \mathbf{J}_{b_2}^t \mathbf{L}_{b_2} \ldots \mathbf{J}_{b_1}^t \mathbf{L}_{b_1}$$

Blend skinning. From per-bone kinematic transformations $\mathbf{T}_b^t$ and root body pose $\mathbf{G}^t$, we use dual-quaternion blend skinning to compute a 3D warping field $\mathcal{W}(\mathbf{X})$:

$$\mathcal{W}(\mathbf{X})^t = (\textstyle\sum_b \mathbf{W}(\mathbf{X})_b^t \cdot \mathbf{T}_b^t) \cdot \mathbf{G}^t$$

Skinning weights $\mathbf{W}(X)_b^t$ are specified by the softmax'ed distances between rest mesh vertices $\mathbf{X}$ and bone centers.

3.4. Network Architecture

Single-frame image encoder. Given a video of length T, we use off-the-shelf PointRend [12] to compute segmentations and DensePose [7, 23, 24] to compute per-pixel CSE features. Each RGB image is concatenated with features, then masked and cropped to 1.2x the tight bounding box of the object. We find that using dense pretrained features improves convergence speed over RGB inputs alone. We pass the preprocessed frames into a convolutional stacked hourglass network [25], which uses repeated pooling and upsampling to process features across multiple scales and spatial locations. The stacked intermediate outputs of each hourglass module are passed into a ResNet18 [8] network which predicts latent vectors z_{view} and z_{art}: z_{view} is decoded into the viewpoint $\mathbf{G}^t \in SE(3)$ and z_{art} is decoded into articulated joint angles $\mathbf{A}^t \in SO(3)^B$. We use 6D rotations to represent angles during training [47].

Viewpoint branch. As the space of possible viewpoints is discontinuous and multi-modal, it is difficult to approach the optimum through iterative gradient descent. Following [6], we use a viewpoint decoder network $\mathbf{MLP}_{\text{view}}$ to decode z_{pose} into a set of M viewpoint hypotheses $\mathbf{G}^t_{\{1,\ldots,M\}}$ weighted by $w^t_{\mathbf{G}} \in \mathbb{R}^M$. Fig. 3 shows the variation over viewpoint hypotheses at an early stage of training.

Articulation branch. Estimating 3D articulations from monocular images can be difficult due to depth ambiguities and occlusions. To resolve this, recent work on human pose estimation [37] uses a normalizing flow articulation prior represented as an invertible neural network, trained on large-scale human motion capture datasets. Without access to such datasets for other categories, we achieve a similar effect by leveraging the teacher's articulation priors, using the teacher's frozen articulation decoder $\mathbf{MLP}_{\text{art}}$ to decode z_{art}. Fig. 4 visualizes the principal components of z_{art}'s latent space: we find that perturbing z_{art} along its principal components causes the resulting articulated shape to perform natural motions, such as curling up in a ball.

Appearance branch. As the object in the video is only partially observable at any given time, we must use data priors or look at nearby frames to output complete appearance predictions $\mathbf{C}^t \in \mathbb{R}^{|V|\times 3}$, represented as per-vertex colors of the articulated mesh at each frame. Our teacher [44] models global object appearances as a category-level neural field modulated by a texture code $z_{\text{color}} \in \mathbb{R}^{64}$. We leverage these appearance priors by predicting an environment code per

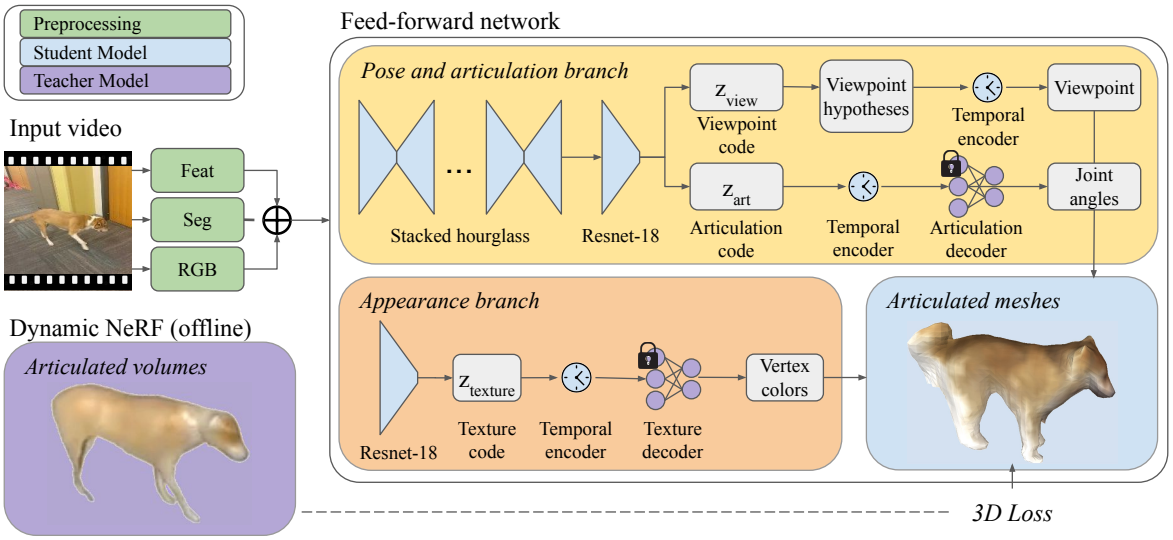

Figure 2. **Architecture details**. We train a feed-forward network (shown in blue) to predict articulations and textures from videos, supervised by the pseudo-ground-truth outputs of an offline dynamic NeRF teacher [44] (including 3D shapes, articulations, and textures). To simplify the learning task, our feed-forward network outputs in a high-dimensional latent space: the dynamic NeRF's frozen decoder networks (shown in purple) are used to convert articulation codes into joint angles and texture codes into 3D surface textures. Spatial per-vertex $\mathcal{L}_2$ losses are used to supervise the articulated 3D outputs against the articulated pseudo-ground-truths.

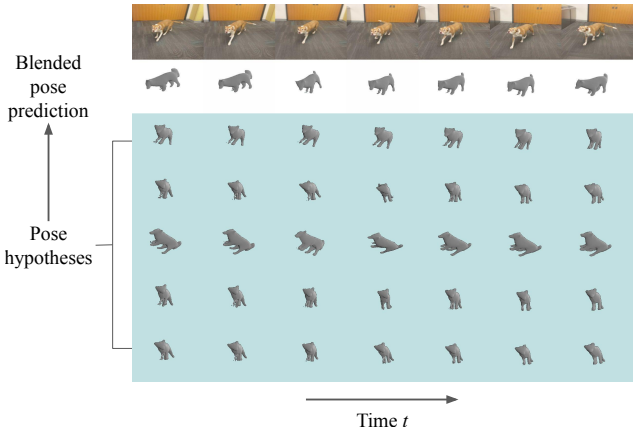

Figure 3. **Viewpoint multiplexing during training**. To overcome the discontinuous and multi-modal nature of pose optimization, we train our feed-forward predictor to output a set of M viewpoint hypotheses rather than a single viewpoint. Blending multiple viewpoint hypotheses yields a more accurate prediction (here $M = 5$). **Top row**: Input frames. **Second row** (white): Blended viewpoint prediction. **Bottom rows** (light blue): M viewpoint hypotheses outputted by single-frame encoder.

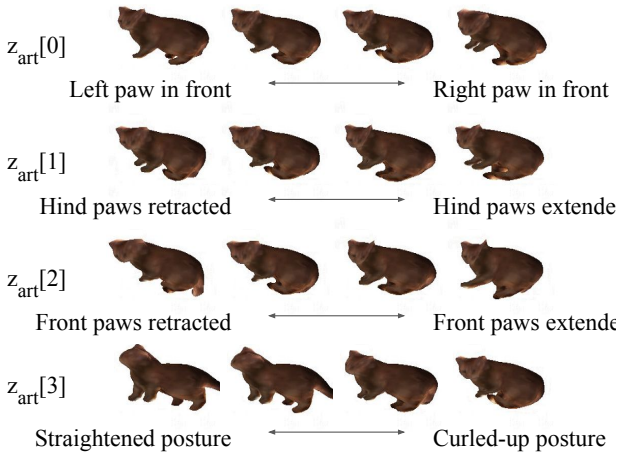

Figure 4. **Visualizing articulation code**. Rather than training our feed-forward predictor to output high-dimensional (75) skeletal articulations, we leverage the teacher's pre-trained articulation space by predicting a low-dimensional (16) code. We visualize the first four principal components of the articulation code learned from the training dataset and find that variations along each axis correspond to interpretable motions: (1) whether the left or right paw is in front, (2) whether the hind paws are retracted or extended, (3) whether the front paws are retracted or extended, and (4) whether the cat has a straightened or curled-up posture.

frame, modulating the teacher's frozen category-level neural field, and querying it at the rest mesh's vertex locations.

Temporal encoder. Predicting pose and shape from single images can be highly ambiguous due to motion blur, occlusions, and depth ambiguities. To output temporally consistent results over long videos, we define a temporal encoder that updates z_{view}, z_{art}, and z_{color} across many frames. We treat the viewpoint multiplex z_{view} as a single vector by concatenating all M hypotheses. Following prior work [11], our temporal encoder has several 1D transformer layers acting on a temporal window centered at time t.

3.5. Losses and Supervision

Optimization objective. Treating the teacher's outputs as pseudo-ground truths, our model can be trained in a standard supervised manner. We would like to use the teacher's inferred low-dimensional latent codes as supervisory targets for the student. But rather than defining $\mathcal{L}_2$ losses in the latent embedding space, where each dimension of z_{view}, z_{art}, and z_{color} has varying importance, we instead render these codes and compute 3D losses. As all articulated meshes have the same topology, we can define geometry and color losses as per-vertex $\mathcal{L}_2$ error.

$$L_{\text{geom}} = \left\| \mathbf{X}^t - \widehat{\mathbf{X}^t} \right\|_2 \qquad L_{\text{color}} = \left\| \mathbf{C}^t - \widehat{\mathbf{C}^t} \right\|_2$$

To improve articulation learning and account for multiple possible inverse kinematics solutions while solving for joint articulations, we add a joint loss defined as geodesic distance between predicted and actual joint angles at each joint. We find that this improves deformation quality.

$$L_{\text{joint}} = D_{\text{geodesic}}(\mathbf{A}^t, \widehat{\mathbf{A}^t})$$

Our training objectives are summarized below:

$$L = L_{\text{geom}} + L_{\text{joint}} + L_{\text{color}}$$

4. Experiments

Although we could have distilled a student using any dynamic NeRF as the teacher, our implementation uses BANMo [44] because it has public code, accepts unannotated monocular videos as input, and produces good results on humans and quadrupeds. We modified BANMo to replace the bag-of-bones deformation model with a skeleton.
Hyperparameters. Our method is implemented in PyTorch. We use the AdamW optimizer and train the model for 16k iterations, taking around 4 hours on a single RTX-3090 GPU. We use 224×224 images with batch size 56. We use 8 stacked hourglass blocks in the image encoder and a window size of 13 frames in the temporal encoder. All losses are weighted to have similar initial magnitudes.
Staged training. We adopt a two-stage training strategy at every epoch to reduce the computational costs of evaluating an image encoder at every frame, when only a single frame per time window will receive a gradient update. In the first stage, we send frames through the image encoder and compute per-frame losses without using the temporal encoder, storing per-frame values of z_{pose}, z_{art}, and z_{color}. In the second stage, we send time windows of cached features through the temporal encoder without using the image encoder. Two-staged training reduces redundant image encoder evaluations and ensures that the temporal encoder can be safely removed if we only have access to single images rather than time windows of frames at test time.

Table 1. **Datasets**. All videos are treated as casually collected monocular RGB videos, except when additional ground-truth meshes or depth maps are needed for evaluation. Processing time includes computing segmentations and per-pixel features with off-the-shelf networks, as well as computing pseudo-ground truth 3D outputs with the teacher.

	Total Videos	Total Instances	Total Frames	Test Instances	Test Frames	Processing Time (hr)
Humans	48	28	6.4k	10	1.6k	22.0
Cats	77	56	11.7k	13	2.7k	23.2
Dogs	88	78	9.7k	17	1.7k	23.8

4.1. Datasets

We collect datasets for three categories: humans, cats, and dogs. For humans, we combine datasets from AMA [35], MonoPerfCap [41], DAVIS [29], and BANMo [44] to obtain 48 human videos. We evaluate on AMA and MonoPerfCap as they have ground-truth meshes. For cats and dogs, we collect 77 cat videos and 88 dog videos from the Pexels stock video website as well as BANMo released data. We also used an iPad Pro to capture two RGB-D videos to evaluate depth accuracy on cats and dogs. Video frames are extracted at 10fps. Datasets are summarized in Tab. 1.

To compute BANMo pseudo-ground-truths per category without optimizing dozens of independent BANMo models, we optimize a *single* BANMo model per category, using articulation and texture codes to model shape and appearance differences across instances. As a result, the total dataset processing time is about 24 hours per category on 8 RTX-3090 GPUs. We find that BANMo generalizes quite well from the instance-level to the category-level setting [45], and that the articulation and texture latent spaces cover the range of motions and textures across all instances.

4.2. Reconstructing Humans

Dataset. Following BANMo, we evaluate human reconstruction on the AMA [35] dataset, containing 10 real-world mesh sequences depicting 3 different humans. The subjects wear loose clothing and perform challenging actions such as dancing and performing a handstand. Although the AMA videos were captured in an 8-camera studio to enable ground-truth mesh extraction, we treat them as casually collected monocular videos and do not use the camera intrinsics, camera extrinsics, or time synchronization.
Comparisons. We compare against template-free BANMo [44], as well as model-based methods HuMoR [31] and ICON [40]. BANMo fits an animatable 3D model to multiple monocular videos of an object instance by performing differentiable rendering optimization. We train BANMo on the same dataset as our model. HuMoR is a human-specific temporal pose and shape predictor that performs test-time optimization on video sequences, leveraging OpenPose keypoint detection and motion priors learned

Figure 5. **Qualitative results on humans (left), cats (middle), and dogs (right) in the test set**. From left to right for each category, we show the input images, articulated shape and texture predictions overlaid on the input images, and three different viewpoints of the predicted geometry. Our method operates on casual monocular videos and predicts plausible shape, articulations, and textures in real-time. Our predictions align well with the image evidence on challenging inputs.

from large-scale human motion capture datasets. ICON is the current SOTA for single-view human reconstruction, and it combines implicit functions with the SMPL human body model, using test-time optimization to fit surface normal predictions and improve pose accuracy and reconstruction quality. All our baselines require far more processing time than our model, which runs in real-time.

Metrics. We report 3D chamfer distance and F-scores on three held-out test sequences in Tab. 2, averaged across all frames. Chamfer distance is the average distance between the ground-truth and predicted mesh vertices using nearest neighbor matches. As this can be sensitive to outliers, we also evaluate F-score at distance thresholds $d \in \{1\%, 2\%, 5\%\}$ to better quantify reconstruction error at different granularities. We scale predicted meshes by their view-space bounding box height to account for unknown scale compared to registered ground-truths. Our model approaches the performance of BANMo and baselines, while requiring nearly 1000x less compute at test time.

Qualitative Results. We show qualitative comparisons for upright and inverted humans in Fig. 7. HuMoR outputs a deformed SMPL model, while ICON and BANMo optimize for both shape and articulation. Although our method outputs reasonable articulations for the upright pose, the woman lacks fine-grained geometry details and the viewpoint on the handstand video is inaccurate. We hypothesize that these failures occur due to the entanglement between articulations and per-instance morphologies, and the lack of handstand examples in the training set.

4.3. Reconstructing Cats and Dogs

Dataset. We evaluate cat and dog reconstruction on two RGB-D pet videos, as well as a held-out test set of pet videos from BANMo's dataset. These videos contain challenging motions such as rapid turns and jumping off chairs.
Comparisons. We compare against BANMo [44] and BARC [32], a model-based approach and the current SOTA for dog shape and pose estimation from images. BARC trains a feed-forward network using images with keypoint labels and synthetic SMAL dog models [4], leveraging breed losses as additional supervision. As BARC is image-based, we run it separately on each video frame.
Metrics We report the root mean square depth error and depth accuracy for all foreground pixels in Tab. 3, averaged across all frames. We render a synthetic depth map per frame, and following [19], we account for the unknown global scale factor between depth maps by aligning the median rendered and ground-truth depths at each frame:

$$s_i = \underset{x}{\text{median}} \left\{ D_i^{pred}(x) / D^{gt}(x) \right\}$$

Depth accuracy is computed as the proportion of foreground pixels whose synthetic depth is within a given threshold.
Qualitative Results We show qualitative results comparing to BARC in Fig. 6. BARC performs well at predicting coarse shape and deformations, and more faithfully captures the fine motion and geometry details of the dog when it is positioned well in frame. However, as BARC entangles shape and breed, we find that BARC may predict biased shapes for certain breeds. For example, in the bottom left

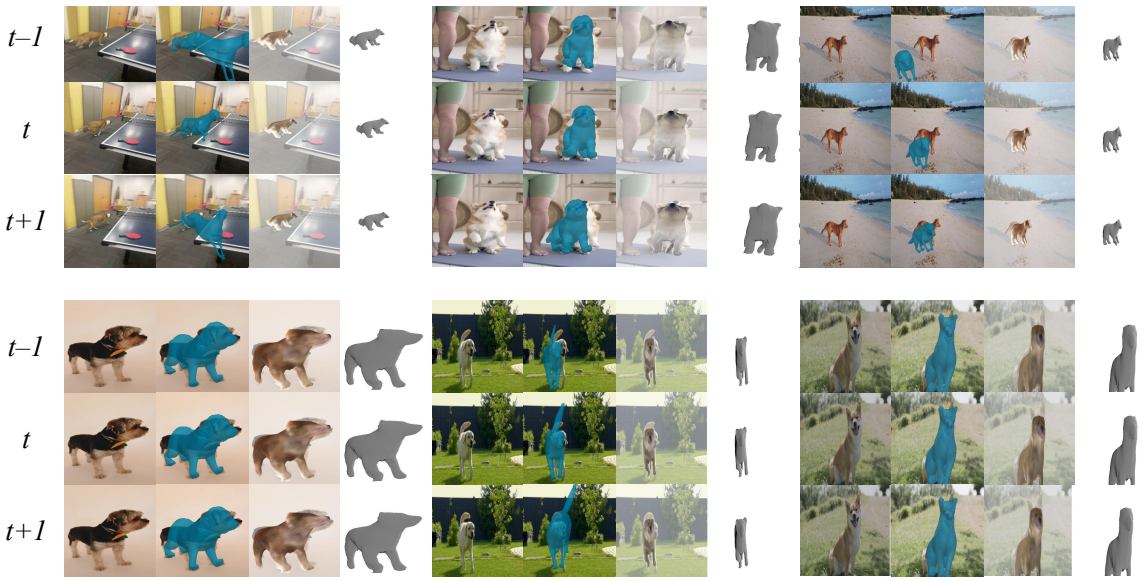

Figure 6. **Qualitative comparisons on pet sequences**. From left to right in each column, we show (**left**) the input image, (**middle-left**) BARC's prediction, (**middle-right**) our articulated shape and texture predictions, and (**right**) our geometry predictions. Our method operates on videos and predicts plausible shape, articulations, and texture in real time. BARC operates on each video frame independently, resulting in jittery predictions when the pet is small or not clearly visible (**top row**). While BARC's geometry and motion predictions are largely accurate, they can be biased for certain breeds (**bottom left**), predicting an arched back for a flat-backed dog.

Table 2. **Quantitative results on AMA sequences.** 3D chamfer distance (cm, ↓) and F-score (%, ↑) for articulated meshes, averaged over all frames. `T_samba` is a held-out video of a person that was seen during training, while the `D_bouncing` and `D_handstand` sequences contain a previously unseen person. Our model approaches the performance of BANMo and baselines while requiring nearly three orders of magnitude less compute at test time. Other baselines are trained on 3D human data, rely on the SMPL body model, or use expensive test-time optimization to improve results. We also report the model inference time (ms) per frame. Results marked by * are different runs with the same hyperparameters. The best results are in bold. Please refer to Sec. 4.4 for discussions on the ablation results.

Method	Time	T_samba				D_bouncing				D_handstand			
		CD	F@1%	F@2%	F@5%	CD	F@1%	F@2%	F@5%	CD	F@1%	F@2%	F@5%
Ours*	**67**	11.17	83.7	61.5	28.8	15.27	73.7	47.2	19.6	24.56	57.0	33.1	13.1
HuMoR	42000	**10.32**	**88.3**	60.8	26.0	11.75	85.1	56.6	23.4	30.24	46.4	25.1	9.7
ICON	63000	10.43	85.9	**62.3**	**29.7**	**9.77**	**88.3**	**65.6**	**31.0**	16.02	72.5	48.2	20.4
BANMo	43000	11.56	82.7	57.0	25.3	10.90	86.2	64.9	29.8	**15.22**	**75.5**	**50.7**	**21.8**
No temporal encoder	**65**	12.86	80.3	56.1	24.6	15.55	73.2	46.9	19.6	27.42	51.1	28.8	11.3
Conv1D encoder	67	12.30	80.4	58.9	28.1	15.92	72.0	44.6	18.4	**23.49**	54.9	31.4	12.4
Transformer encoder*	72	**11.49**	**83.2**	**61.1**	**28.9**	**14.47**	**76.9**	**51.5**	**21.8**	27.15	**55.6**	**36.2**	**15.4**
w/o frozen decoders	66	14.10	77.7	51.2	21.9	14.72	75.7	48.6	19.9	31.15	49.7	31.5	13.1
64^3 template grid	55	14.21	76.9	51.6	21.6	16.27	72.0	45.9	18.8	24.31	52.9	29.0	11.2
128^3 template grid*	**67**	**10.84**	**85.0**	**62.4**	**29.6**	**15.10**	**75.1**	**48.7**	**20.3**	25.70	**56.0**	**35.8**	**15.2**
256^3 template grid	73	12.38	80.7	58.0	27.0	15.60	73.6	47.7	20.0	**23.99**	54.5	31.3	12.7
5 train videos	**67**	46.98	22.1	10.4	4.2	18.45	65.2	39.0	15.6	**23.72**	**54.6**	**31.6**	**13.0**
16 train videos	**67**	34.23	32.5	17.6	7.1	16.87	69.8	42.9	17.7	24.67	54.4	29.7	11.3
26 train videos	**67**	**10.92**	**83.8**	**62.8**	**32.0**	**16.04**	**72.5**	**46.9**	**19.5**	28.07	47.5	26.1	10.2
35 train videos*	**67**	12.28	81.4	57.3	25.7	16.58	70.4	42.5	17.2	27.64	42.2	22.5	8.9

of Fig. 6, BARC predicts a rounded back but the back is flat in reality. For the inputs on the top row, BARC's single-frame architecture makes jittery predictions from frame to frame, while our temporal architecture enforces consistency and prevents large discontinuities in pose and deformation. In the middle top image, which is particularly challeng-

Table 3. **Quantitative results on RGB-D pet sequences.** Root mean square depth error (↓) and depth accuracy (%, ↑) for all foreground pixels in the depth map, averaged over all frames. We also report the model inference time per frame (ms) on a RTX-3090 GPU. Although BANMo consistently does better, our method approaches its quality while being nearly 1000x faster at test time. Best results are in bold.

Method	Time	dog				cat			
		RMSE	Acc-1%	Acc-2%	Acc-5%	RMSE	Acc-1%	Acc-2%	Acc-5%
BANMo	54000	**0.0411**	**28.6**	**45.3**	**72.7**	**0.0757**	**27.7**	**47.4**	**76.5**
Ours	**72**	0.0621	13.4	24.2	46.2	0.1292	20.9	35.0	63.7

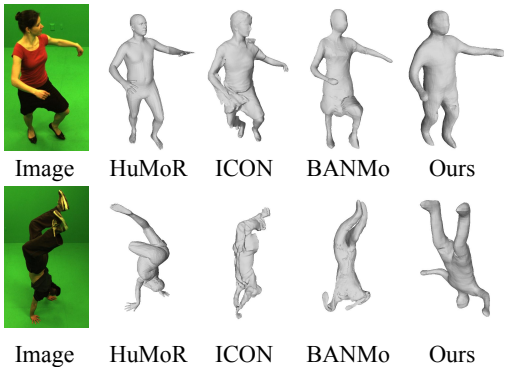

Figure 7. **Qualitative comparisons on human sequences**. From left to right, we show the output of HuMoR [31], ICON [40], BANMo [44], and our method on dancing (**top**) and handstand (**bottom**) sequences. Our method outputs plausible articulations for the dancing sequence, although the shape lacks fine details. In comparison, HuMoR outputs the SMPL template human shape, while ICON and BANMo output more detailed shapes. All methods perform poorly on the highly challenging handstand pose: HuMoR outputs unrealistic and self-intersecting articulations. ICON's prediction appears bumpy and twisted in an unnatural way. BANMo's prediction is also twisted and misses the head. Our method outputs incorrect viewpoint and arm articulations.

ing because the dog is standing on its hind legs, both our method and BARC fail to show that the dog is lifting its front paws. As our method does not disentangle articulation and morphology variation between breeds, incorporating breed and/or instance information would likely improve our ability to represent fine motion and geometry details.

4.4. Ablations

We performed an ablation study on several architectural details including the choice of temporal encoder, the use of frozen decoders from the teacher, the template grid resolution, and the number of training videos. Most notably, we find that increasing the density of mesh points by running marching cubes at a finer resolution (from 64^3 to 128^3 improves the `F@1%` and `F@2%` accuracy of our student, as well as using a temporal encoder. Changing the marching cubes resolution involves no change to the student, but simply a more faithful post-processing of the teacher's output.

Effect of temporal encoder. The student model performs better with a temporal encoder across all sequences, perhaps by providing additional temporal context to help smooth the predictions. In general, the transformer encoder seems to outperform the 1D convolutional encoder in accuracy but not speed.

Effect of frozen decoders. We find that the student model's performance generally drops without using the teacher's frozen articulation decoder to regularize pose predictions.

Effect of template shape resolution. When extracting the template shape with a 64^3 grid, our model's `F@1%`, `F@2%`, and `F@5%` accuracy all seem to drop. Qualitatively, the 64^3 template shape is less detailed and appears fuzzy compared to 128^3 or 256^3, though the predicted articulations are similar. There seems to be little difference between 128^3 and 256^3, and the observed results could be up to noise. As more points need to be warped to the deformed space, more detailed templates have slower inference speed.

Effect of dataset size. We find that performance generally deteriorates with less data, although the trend is not as clear. Here, each smaller set of training videos is a subset of the immediately larger video set. As all videos containing the `T_samba` identity are within the set of 26 videos, increasing the dataset size from 26 to 35 involved adding videos of unrelated identities which seemed to hurt performance.

5. Discussion

We present a method to train category-specific feed-forward video shape predictors by distilling knowledge from dynamic NeRF teachers fitted to offline video data at scale. Our temporal architecture predicts consistent viewpoint, articulation, and appearance, producing real-time video reconstruction results on humans, cats, and dogs with the ability to support other categories as well. We qualitatively outperform existing feed-forward predictors for dog shape and pose, and approach the quality of test-time fitting methods while using nearly 1000x less computation.

Limitations: As our method is trained on the pseudo-ground truth outputs of a teacher model, we are upper-bounded by the teacher's performance and reconstruction fidelity. We expect the performance of our method to improve given larger-scale, more diverse, and higher-quality video data. Compared with optimization-based methods, our method is hundreds of times faster but produces results less faithful to the inputs: we leave incorporating optimization into the feed-forward architecture as future work.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *ICCV*, 2009. 1

[2] Marc Bager, Yufu Wang, Adarsh Modh, Ammon Perkes, Nikos Kolotouros, Bernd Pfrommer, Marc Schmidt, and Kostas Daniilidis. 3d bird reconstruction: A dataset, model, and shape recovery from a single view. *ECCV*, 2020. 2

[3] Benjamin Biggs, Ollie Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. Who Let the Dogs Out: 3D Animal Reconstruction with Expectation Maximization in the Loop. *ECCV*, 2020. 2

[4] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures Great and SMAL: Recovering the Shape and Motion of Animals From Video. *ACCV*, 2018. 1, 6

[5] Tretschk Edgar, Ayush Teawri, Vladislav Golyanik, Michael Zollhofer, Christoph Lassner, and Christian Theobalt. Non-rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene from Monocular Video. *ICCV*, 2021. 2

[6] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and Viewpoints Without Keypoints. *ECCV*, 2020. 2, 3

[7] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense Human Pose Estimation In The Wild. *CVPR*, 2018. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv*, 2015. 3

[9] Alec Jacobson and Olga Sorkine. Stretchable and Twistable Bones for Skeletal Shape Deformation. *SIGGRAPH Asia*, 2011. 3

[10] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning Category-Specific Mesh Reconstruction from Image Collections. *ECCV*, 2018. 2

[11] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3D Human Dynamics from Video. *CVPR*, 2019. 4

[12] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image Segmentation as Rendering. *CVPR*, 2020. 3

[13] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Video Inference for Human Body Pose and Shape Estimation. *CVPR*, 2020. 1, 2

[14] Filippos Kokkinos and Iasonas Kokkinos. To the Point: Correspondence-Driven Monocular 3D Category Reconstruction. *NeurIPS*, 2021. 2

[15] Binh Huy Le and Zhigang Deng. Robust and Accurate Skeletal Rigging from Mesh Sequences. *ACM TOG*, 2014. 3

[16] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, and Angjoo Kanazawa. TAVA: Template-free Animatable Volumetric Actors. *CVPR*, 2022. 1, 2

[17] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *CVPR*, 2021. 2

[18] Matthew Loper, Naureen Mahmood, Javier Romero, gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. *SIGGRAPH Asia*, 2015. 1, 2

[19] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent Video Depth Estimation. *SIGGRAPH*, 2020. 6

[20] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild. *CVPR*, 2021. 1, 2

[21] 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. *ECCV*, 2020. 1, 2

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *SIGGRAPH*, 2022. 2

[23] Natalia Neverova, David Novotny, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous Surface Embeddings. *NeurIPS*, 2020. 3

[24] Natalia Neverova, Artsiom Sanakoyeu, Patrick Labatut, David Novotny, and Andrea Vedaldi. Discovering Relationships Between Object Categories via Universal Canonical Maps. *CVPR*, 2021. 3

[25] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. *ECCV*, 2016. 3

[26] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch It Move: Unsupervised Discovery of 3D Joints for Re-posing of Articulated Objects. *CVPR*, 2021. 3

[27] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. *ICCV*, 2021. 1, 2

[28] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *SIGGRAPH Asia*, 2021. 1

[29] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. *CVPR*, 2016. 5

[30] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *CVPR*, 2020. 1, 2

[31] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. HuMoR: 3D Human Motion Model for Robust Pose Estimation. *ICCV*, 2021. 1, 5, 8

[32] Nadine Rueegg, Silvia Zuffi, Konrad Schindler, and Michael J. Black. BARC: Learning to Regress 3D Dog Shape from Images by Exploiting Breed Information. *CVPR*, 2022. 2, 6

[33] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo Tourism: Exploring Photo Collections in 3D. *SIGGRAPH*, 2006. 1

[34] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit Mesh Reconstruction From Unannotated Image Collections. *arXiv*, 2020. 2

[35] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic. Articulated Mesh Animation from Multi-View Silhouettes. *ACM TOG*, 2008. 5

[36] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural Radiance Fields Without Known Camera Parameters. *arXiv*, 2021. 1, 2

[37] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic Monocular 3D Human Pose Estimation with Normalizing Flows. *ICCV*, 2021. 3

[38] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. DOVE: Learning Deformable 3D Objects by Watching Videos. *arXiv*, 2022. 2

[39] Yuefan Wu, Zeyuan Chen, Shaowei Liu, Zhongzheng Ren, and Shenlong Wang. CASA: Category-Agnostic Skeletal Animal Reconstruction. *CVPR*, 2019. 3

[40] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed Humans Obtained from Normals. *CVPR*, 2022. 5, 8

[41] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. MonoPerfCap: Human Performance Capture from Monocular Video. *ACM TOG*, 2018. 5

[42] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T. Freeman, and Ce Liu. LASR: Learning Articulated Shape Reconstruction from a Monocular Video. *CVPR*, 2021. 2

[43] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. ViSER: Video-Specific Surface Embeddings for Articulated 3D Shape Reconstruction. *NeurIPS*, 2021. 2

[44] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. BANMo: Building Animatable 3D Neural Models from Many Casual Videos. *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 8

[45] Gengshan Yang, Chaoyang Wang, N Dinesh Reddy, and Deva Ramanan. RAC: Reconstructing Animatable Categories from Videos. *CVPR*, 2023. 5

[46] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields Without Neural Networks. *CVPR*, 2022. 2

[47] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. *CVPR*, 2019. 3

[48] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and Tigers and Bears: Capturing Non-Rigid, 3D, Articulated Shape From Images. *CVPR*, 2018. 2

[49] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. *CVPR*, 2017. 2