Autoregressive Appearance Prediction for 3D Gaussian Avatars

Michael Steiner[1,2⋆], Zhang Chen[2], Alexander Richard[2], Vasu Agrawal[2], Markus Steinberger[1], and Michael Zollhoefer[2]

[1] Graz University of Technology, Austria
[2] Meta Reality Labs, Pittsburgh, USA
https://steimich96.github.io/AAP-3DGA/

Abstract. A photorealistic and immersive human avatar experience demands capturing fine, person-specific details such as cloth and hair dynamics, subtle facial expressions, and characteristic motion patterns. Achieving this requires large, high-quality datasets, which often introduce ambiguities and spurious correlations when very similar poses correspond to different appearances. Models that fit these details during training can overfit and produce unstable, abrupt appearance changes for novel poses. We propose a 3D Gaussian Splatting avatar model with a spatial MLP backbone that is conditioned on both pose and an appearance latent. The latent is learned during training by an encoder, yielding a compact representation that improves reconstruction quality and helps disambiguate pose-driven renderings. At driving time, our predictor autoregressively infers the latent, producing temporally smooth appearance evolution and improved stability. Overall, our method delivers a robust and practical path to high-fidelity, stable avatar driving.

Keywords: Human Avatars · Reconstruction · Animation

1 Introduction

Learning a high-fidelity, photorealistic virtual representation of a person has many applications for immersive experiences, such as virtual reality (VR), telepresence, video games and movies. However, producing a person-specific avatar that captures fine details faithfully—*e.g.*, cloth and loose-hair dynamics, subtle facial expressions, and characteristic motion patterns—typically requires extensive multi-view capture data. Long-form captures often exhibit a pose–appearance ambiguity: the same skeletal pose can correspond to noticeably different appearances at different times. This can happen, for example, when clothing is readjusted, hair settles differently, or wrinkles form in a new pattern. A robust avatar model must simultaneously (i) represent high-frequency, view-dependent, and non-rigid effects, and (ii) remain temporally stable when the driving signal revisits similar poses. Recent progress in 3D Gaussian Splatting (3DGS) [18] has

⋆ Work done during an internship at Meta

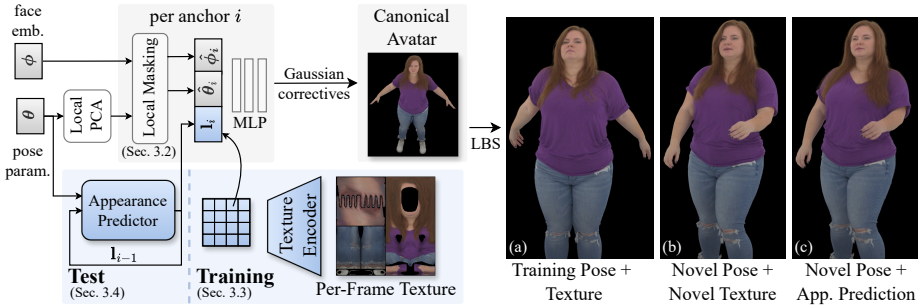


Fig. 1: We represent the posed avatar with a hierarchical 3D Gaussian structure controlled by per-anchor spatial MLPs. Each anchor receives localized driving features via skinning-weight-based masking of pose parameters $\boldsymbol{\theta}$ (and face-region masking of $\boldsymbol{\phi}$), together with an appearance latent $\mathbf{l}_i$. During training, $\mathbf{l}_i$ is obtained by encoding a per-frame UV texture into a 2D feature map and sampling it at the anchor's UV coordinates; at test time, a transformer autoregressively predicts temporally smooth latents from a short pose history for stable driving. The encoder reconstructs training poses extremely well (a) and generalizes to novel poses and unseen textures (b), while our appearance predictor yields realistic appearances and smooth transitions on test sequences when textures are not available (c).

made it an attractive representation for human avatars. Its point-based structure provides the flexibility to model complex deformations, while enabling high-fidelity reconstructions that capture view-dependent effects and thin structures such as hair. Yet, increasing model capacity introduces a second, closely related challenge: spurious correlations in the training data. For instance, a particular hand configuration may be correlated with a specific wrinkle pattern on the trousers, even though it cannot cause that appearance change. High-capacity models can exploit such correlations and memorize incidental details, causing erratic appearance changes and temporal flickering for novel pose sequences.

We propose a 3DGS-based avatar model that addresses both **spurious pose correlations** and **pose–appearance ambiguity** while maintaining high visual fidelity, dynamic effects, and smooth appearance transitions. Our method is built on a hierarchical point representation [46]: a sparse set of anchors with spatially distributed MLPs, control points that guide positional displacement, and a dense set of 3D Gaussians that represent the remaining appearance attributes. To reduce spurious long-range dependencies between unrelated pose parameters, we condition each spatial MLP only on **localized pose** information derived from skinning weights at its anchor location. While this locality improves causal consistency, it also makes the pose–appearance ambiguity more pronounced: local pose parameters alone are often insufficient to determine the correct time-varying appearance. To resolve this ambiguity, we introduce a per-frame appearance latent code learned during **training** via an **appearance encoder**. The encoder takes as input a UV texture obtained by multi-view projection onto the tem-

plate mesh, providing a compact, pose-aligned appearance observation without the need for computationally expensive per-frame mesh registration. Because this UV texture is typically not available at **test time**, we train an additional transformer-based **appearance predictor** after avatar training to regress these encoder-produced latent codes. Given a window of previous poses and the previous latent code, it predicts the next latent code, enabling temporally smooth and realistic appearance evolution during driving.

Combined, these components allow our model to fit challenging captures at exceptionally high fidelity while producing stable and plausible appearance transitions for novel pose sequences. We validate our approach in quantitative and qualitative experiments on six extensive, high-quality captures, showing significant improvements in temporal stability and adherence to the driving signal.

In summary, our contributions are:

- A spatial-MLP-conditioned 3D Gaussian avatar model that combines localized pose conditioning with per-frame appearance latents to handle pose–appearance ambiguity and spurious pose correlations.
- A practical appearance encoding scheme that enables fast and stable learning of per-frame appearance latent codes during training.
- A transformer-based appearance predictor that generates temporally smooth appearance latents at test time from pose history.

2 Preliminaries & Related Work

As our work focuses on animatable, high-fidelity, and faithful reconstruction of person-specific avatars, we will not cover the large body of work on replay [14, 26, 34, 39] and monocular or single-image reconstruction [6, 10–12, 15, 16, 51].

Mesh-based: Mesh-based representations are widely used for modeling human avatars. Early approaches either simulated clothing dynamics explicitly [8] or synthesized novel views and motions via retrieval and interpolation from the captured dataset [2, 43]. To reduce storage and improve fidelity for unseen poses, later works introduced neural networks to compute pose-dependent textures [9]. Several methods further factorize the problem by modeling garments separately [40, 41], through inverse physics [49], and/or incorporate depth cues to improve drivability and robustness [42]. Mesh pipelines have also been combined with deferred shading and neural rendering modules [28], or generative models [48]. To better follow driving signals and reduce entanglement between motion and appearance, Bagautdinov et al. [1] propose separating pose and appearance factors and enforcing disentanglement via mutual information minimization.

NeRF-based: Following NeRF [29], many avatar methods adopted volumetric rendering to better capture view-dependent effects and fine detail. Most are template-based—either relying on a fitted template mesh [9] or learning deformations of a canonical space [25, 33, 44, 50]—with template-free variants also

explored [22]. Although advancements in NeRFs improved its practicality via efficient encodings [31] and factorized representations [3], real-time deployment—especially on low-power devices—remains challenging due to the multi-sample ray marching required for high quality, motivating more explicit alternatives.

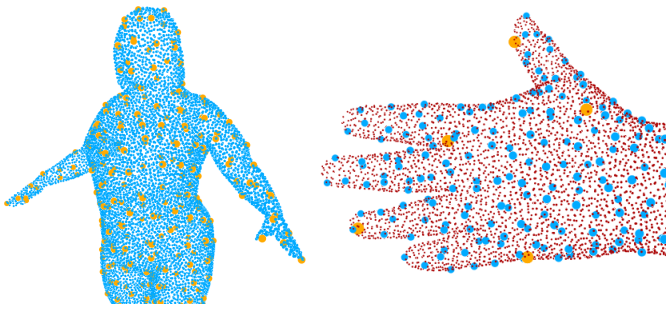


Fig. 2: Following Zhan et al. [46], we initialize a hierarchical point cloud on the template mesh, consisting of anchors/control points/Gaussians (300/10k/200k for our example, colored as orange/blue/red). Every anchor holds an MLP, whose outputs are interpolated by control points and Gaussians from the closest three anchors to calculate their positional displacement and Gaussian correctives.

3DGS-based: Concurrently to NeRF-based methods, point-based avatar representations [24, 27] emerged, offering flexible modeling of complex deformations and convenient manipulation due to the independence of individual primitives. 3D Gaussian Splatting (3DGS) [18] largely superseded earlier point-based and many NeRF-based formulations by combining point-based flexibility with smooth spatial falloffs and favorable optimization behavior. Building on 3DGS, early avatar methods focused primarily on geometric reconstruction and animation, often with limited pose-dependent appearance [20, 21]. Subsequent work introduced stronger pose-conditioned deformation and shading models, e.g., via large MLPs [30, 35] or CNN-based architectures [5, 17, 32]. Other directions target improved drivability and cloth behavior [52], relightability [37], or efficiency for mobile and VR rendering [4, 13, 36]. Animatable Gaussians [23], predicts front–back UV maps with a single large CNN and achieves high visual quality, but falls short of real-time requirements. To improve efficiency without sacrificing capacity, Zhan et al. [46] propose a hierarchical representation consisting of anchors, control points, and Gaussians, where each anchor holds a small spatially localized MLP (visualized in Fig. 2). The MLP outputs are interpolated and combined with a per-Gaussian linear basis to produce Gaussian property correctives; positional updates are instead propagated from control points to enforce smooth deformations (see supplementary for details). This design combines computationally cheap local MLPs with high representational power and well-

behaved deformations, providing a compelling alternative to monolithic MLP- or CNN-based predictors.

3 Method

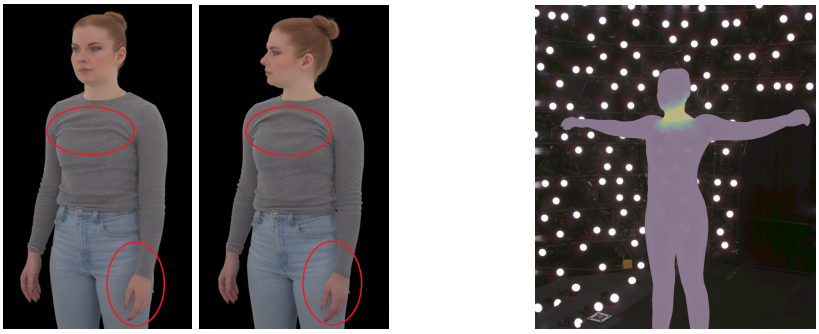
This section details our avatar model and driving pipeline. We begin with a high-level overview of the representation and conditioning strategy (Sec. 3.1). We then describe how we construct *localized* pose features at each anchor to mitigate spurious pose–appearance correlations and reduce overfitting (Sec. 3.2). Next, we introduce our appearance modeling approach, including the per-frame latent learned during training and the encoder used to obtain it (Sec. 3.3). Finally, we present the transformer-based appearance predictor used at test time to autoregressively infer latents for driving.(Sec. 3.4).

3.1 Overview

Given driving pose parameters $\boldsymbol{\theta}$ [7] and face embeddings $\boldsymbol{\phi}$, our model outputs a posed avatar represented as a set of 3D Gaussians. We follow the hierarchical representation of [46], using a sparse set of anchor points with spatially distributed MLPs that predict Gaussian parameters from per-anchor inputs (*cf*. Fig. 2 for a visualization). All points are initialized on the template mesh, receiving transferred skinning weights for linear blend skinning (LBS) and UV coordinates for texture-space lookups. To reduce spurious long-range pose correlations, each anchor i receives **localized** driving features: we apply skinning-weight-based **local masking** to obtain $\hat{\boldsymbol{\theta}}_i$, and mask $\boldsymbol{\phi}$ outside the face UV region to obtain $\hat{\boldsymbol{\phi}}_i$. To further stabilize driving, we constrain poses using PCA computed from the training set [23]; importantly, we apply this transform during both training and test time. Because a single global PCA entangles distant pose parameters, we perform **local PCA** separately over seven body regions.

Finally, we explicitly **model appearance** with per-frame local latent codes. During training, we encode a per-frame UV texture—extracted via multi-view projection onto the template mesh—with a convolutional appearance encoder into a 2D feature map, and bilinearly sample it using each anchor's UV coordinates to obtain $\mathbf{l}_i$. At test time, when UV textures are typically unavailable, we replace the encoder with a transformer **appearance predictor** that regresses the next latents from a short history of masked poses (and the previous latent), enabling smooth and stable appearance evolution during driving. A full overview of our method can be seen in Fig. 1.

Losses. We train our model with a number of commonly used losses for human avatar reconstruction. Specifically, we use a photometric ℓ_1 loss $\mathcal{L}_1$, a perceptual loss $\mathcal{L}_{\text{lpips}}$ [38], an opacity regularizer $\mathcal{L}_{\text{opac}}$ that encourages opaqueness in non-boundary regions, and a scale regularizer $\mathcal{L}_{\text{scale}}$ to discourage overly large Gaussians. Following [46], we enforce smooth deformations by penalizing differences between each control point's displacement and that of its five neighbors,



(a) Spurious correlation of the neck parameter with unrelated regions (*e.g.*, wrinkles and arm's shadow)

(b) Region of influence for the neck twist pose parameter, visualized per Gaussian.

Fig. 3: (a) Providing pose parameter to every anchor leads to spurious correlations between unrelated regions. (b) We locally mask out pose parameters per anchor based on the skinning weights of the template mesh to restrict them to their local region.

yielding $\mathcal{L}_{\text{cpt}}$. Finally, we encourage a well-behaved and smooth appearance latent space via a Kullback–Leibler divergence penalty $\mathcal{L}_{\text{KL}}$. Additional training details are provided in the supplementary material. The total loss is

$$\mathcal{L} = \mathcal{L}_1 + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}} + \lambda_{\text{opac}}\mathcal{L}_{\text{opac}} + \lambda_{\text{scale}}\mathcal{L}_{\text{scale}} + \lambda_{\text{cpt}}\mathcal{L}_{\text{cpt}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}. \quad (1)$$

3.2 Localized Pose Parameters

Conditioning every anchor on the full pose vector can induce spurious correlations between parameters at distant surface regions—*e.g.*, a head rotation becoming correlated with a particular shirt wrinkle pattern (Fig. 3a). Such shortcuts are easy for high-capacity decoders to memorize and often manifest as abrupt appearance changes under novel driving motions. Since non-rigid deformations and cloth dynamics are predominantly local, we introduce a pose-localization scheme that limits each anchor's receptive field in pose space.

Localized Masking. We restrict each pose parameter to the surface regions it can plausibly influence by masking the global pose vector $\boldsymbol{\theta} \in \mathbb{R}^P$ per anchor. For an anchor i on the template mesh, we trace the connection from its skinning weights to the pose parameters and construct a binary mask $\mathbf{m}_i^\theta \in \{0,1\}^P$, where $\mathbf{m}_i^\theta[j]=1$ iff any non-zero skinning weight affects $\boldsymbol{\theta}[j]$. An example mask is shown in Fig. 3b. Similarly, we construct a mask $m_i^\phi \in \{0,1\}$ that is set if the anchor's texture coordinates lie inside the face region. The anchor MLP then receives the localized pose and face embeddings

$$\hat{\boldsymbol{\theta}}_i = \mathbf{m}_i^\theta \odot \boldsymbol{\theta}, \quad \hat{\boldsymbol{\phi}}_i = m_i^\phi \boldsymbol{\phi}, \quad (2)$$

instead of the full $\boldsymbol{\theta}, \boldsymbol{\phi}$. In practice, skinning-weight locality is well suited for tight clothing, but can be overly restrictive for looser garments whose motion

may couple across larger regions. To account for this, we dilate $\mathbf{m}_i^\theta$ once by propagating active entries to its neighboring anchors. Finally, because finger pose parameters can have very small spatial support and may not cover any anchors, we reuse the wrist mask for the corresponding hand and finger parameters.

Localized PCA. Driving with poses far outside the training distribution may produces severe artifacts. Following prior work [23, 46], we therefore constrain poses using a PCA model fit to all training poses: we transform a pose into PCA space, clamp coefficients to $\pm 2\sigma$, and apply the inverse transform. Unlike previous approaches, we apply the same PCA transform and clamping during *both* training and test time, avoiding a train–test distribution mismatch.

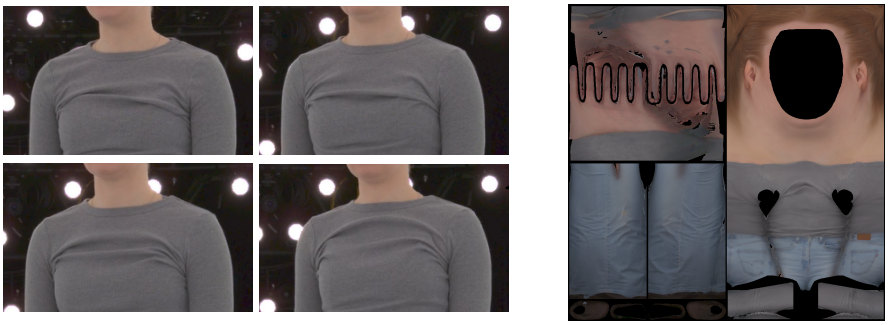
A drawback of global PCA is that it directly entangles distant pose parameters during training, which can reintroduce long-range correlations even in the presence of local masking. To mitigate this, we perform **localized PCA**: we partition pose parameters into seven body-part groups and run PCA independently per group, using 5 PCA components each. The groups are torso+head (12 pose parameters), left/right leg (9 each), left/right arm (12 each), and left/right hand (27 each). This preserves correlations within a body part while preventing global entanglement across unrelated regions.

3.3 Appearance Modeling

Long capture sequences—particularly with rigid garments or loose hair—often exhibit a pronounced pose–appearance ambiguity: nearly identical poses can correspond to noticeably different appearance (*e.g.*, hair settling differently or new wrinkle configurations when returning to a neutral stance; see Fig. 4a). When an avatar is driven by pose alone, a high-capacity decoder can memorize these incidental correlations, leading to abrupt switching between appearance states. To explicitly decouple pose from time-varying appearance, we introduce a learnable per-frame appearance latent.

A naïve per-frame codebook is data-inefficient, typically requires additional temporal regularization to yield a coherent latent space, and converges slowly. Instead, we learn the latent through a convolutional texture encoder that takes as input a per-frame UV texture extracted by multi-view projection onto the template mesh (see Fig. 4b and the supplementary for details). While this projection is imperfect due to small misalignments between the template mesh and the posed surface, it provides a sufficiently stable, pose-aligned reference that accelerates convergence and yields meaningful appearance codes without requiring computationally expensive mesh registration and texture fitting. Since facial detail and expression should be explained by the face embedding $\boldsymbol{\phi}$, we mask the face region in the UV texture before encoding.

The encoder maps a 1024×1024 UV texture to a low-resolution feature map of size $32\times32\times N_l$. For each anchor, we bilinearly sample this map at the anchor's UV coordinate to obtain a local latent code $\mathbf{l}_i$. Compared to global latents, spatially varying codes better match the locality of non-rigid appearance changes, while bilinear sampling enforces smooth transitions across neighboring anchors.



(a) Different appearance (folding) on very similar poses. **(b)** Input to the appearance encoder.

Fig. 4: (a) Similar poses can exhibit vastly different appearances, leading to an ambiguous one-to-many mapping. Therefore, we model appearance through a texture encoder that receives (b) a multi-view projected UV texture as input (face region masked).

3.4 Appearance Prediction

To drive the avatar at test time without UV texture inputs, we learn an appearance predictor that infers the per-anchor appearance latents from motion history. We model latent dynamics with a causal autoregressive transformer that takes the previous N_b body poses and the previous-frame appearance codes as context, and predicts the appearance codes for the current frame (see Fig. 5). Unless stated otherwise, we use N_b=10, which provides sufficient temporal context while generalizing well to unseen sequences.

Naïvely concatenating all pose and latent dimensions into a single vector per frame led to overfitting in our experiments. Instead, we represent the input as a set of tokens, treating each local pose parameter and each local appearance code as its own token. For pose tokens, we use an extended representation consisting of *value*, *velocity*, and *acceleration*, where velocity and acceleration are computed via finite differences. We additionally normalize each training input sequence with respect to its last frame by removing that frame's global translation and rotation from all poses in the input sequence.

The model uses separate lightweight modality encoders to project pose and appearance latent codes into a common token space: a dedicated shallow MLP encoder for pose tokens and another for appearance tokens. We add learnable positional embeddings before the modality encoders, and feed the resulting tokens through a stack of self-attention transformer blocks. The output tokens represent the current frame's local appearance codes.

Training. At inference, the model must bootstrap without an existing historical context. Initializing the context with zeros can create a train–test mismatch and destabilize autoregressive decoding. To close this gap, we train with dual-context initialization: for each training sample we forward the model twice—(i) with ground-truth history appearance codes and (ii) with zeroed codes—and sum the losses from both predictions. This enables recovery from zero initialization while

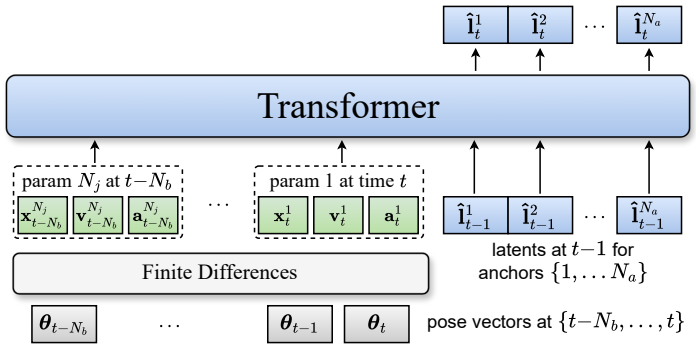


Fig. 5: Architecture of our appearance predictor. The transformer predicts the next latents from the previous N_b poses—value, plus their velocity and acceleration calculated via finite differences—and the previous latents.

maintaining accuracy when a valid history exists. Autoregressive decoding can also suffer from error accumulation over long sequences. To improve robustness, we explicitly unroll the autoregressive predictions over the input window during training—*i.e.*, feeding back the model's prediction at time $t-1$ as input for time t—and we accumulate losses across the unrolled window. Finally, we apply PCA with clamping (as in Sec. 3.2) to the predicted appearance codes during both training and testing, ensuring outputs remain within the training latent distribution and reducing collapse and visual artifacts.

Losses. We supervise appearance prediction with an ℓ_1 loss on the latent codes for both the ground-truth- and zero-initialized prediction. To encourage temporal stability, we additionally apply ℓ_1 penalties to finite-difference derivatives of the velocity, acceleration, and jerk of the latent code output. All loss terms are weighted equally (weight 1).

4 Evaluation

We evaluate our method on six extensive long-form dome captures spanning a range of difficulty and dynamic appearance effects. We compare against a re-implementation of Zhan et al. [46], augmented with our localized face embedding and localized hand/finger pose conditioning to allow for a fair comparison; we denote this baseline as *MMLPs*†. Additionally, we compare against a convolutional decoder baseline similar to a non-relightable version of the Relightable Full-Body Gaussian Codec Avatar (RFGCA) model from Want et al. [37]—denoted as *nRFGCA*†. All models are implemented in PyTorch and use *gsplat* [45] for differentiable 3D Gaussian rendering.

Dataset. Our dataset comprises six dome captures, each with ∼30k training frames at ∼30 fps and ∼250 synchronized training cameras, split evenly between

Table 1: Image metrics on all compared methods. As appearance is inherently ambiguous and drifts over time, we initialize the appearance predictor for Ours with the encoder output. Additionally, we show results for re-initializing every 30 frames, and for using the encoder during driving.

	Test			Train			
	PSNR$^\uparrow$	SSIM$^\uparrow$	LPIPS$^\downarrow$	PSNR$^\uparrow$	SSIM$^\uparrow$	LPIPS$^\downarrow$	#Params
Ours	32.99	0.939	0.063	36.21	0.950	0.053	192.1M
MMLPs†	32.34	0.937	0.066	36.13	0.950	0.054	187.2M
nRFGCA†	32.74	0.939	0.065	35.94	0.950	0.055	155.9M
Ours (re-init. 30)	33.85	0.942	0.060	–	–	–	192.1M
Ours (encoder)	34.78	0.946	0.056	–	–	–	189.8M

full-body, upper-body, and lower-body views. Each capture contains both instructed motion sequences and less scripted segments. All images (with corresponding foreground segmentation masks) for training and evaluation are rendered at a resolution of 1152×1332 ($4\times$ downscaled from the captured images). For evaluation, we hold out $\sim$3k frames and 6 cameras during training (four full-body cameras, plus one frontal upper-body and one frontal lower-body camera).

Training Details. We train and evaluate all methods on 8 NVIDIA H200 GPUs with a total batch size of 64 for 100k iterations, resulting in 8–10 hours of training time. Following [46], *Ours* and *MMLPs*† use 300/10k/200k points for anchors/control points/Gaussians. Our appearance model uses N_l=16 latent channels. We optimize with AdamW using ϵ=10^{-15}, β_1=0.9, and β_2=0.999, and apply weight decay only to the spatial MLPs and the appearance encoder, with λ=0.001. The loss weights are set to λ_{lpips}=0.1, λ_{lpips}=0.1, λ_{opac}=0.5, λ_{scale}=1.0, λ_{cpt}=0.5, λ_{KL}=10^{-6}.

4.1 Results

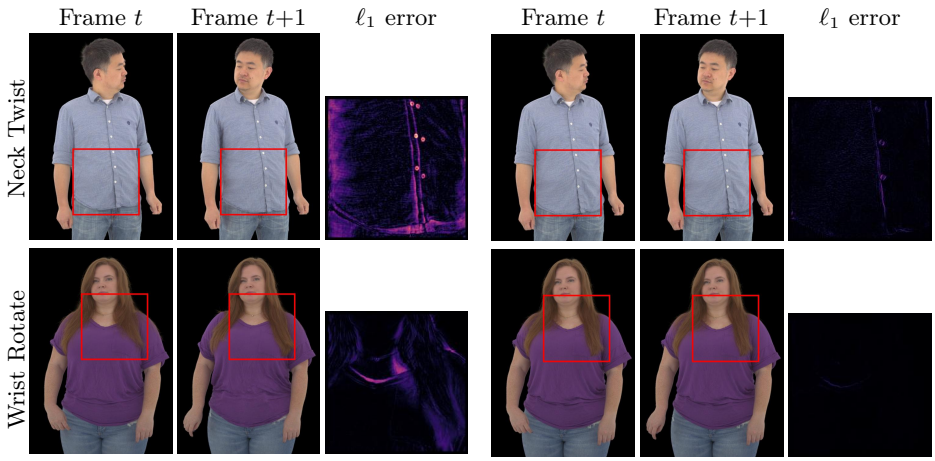
We present quantitative and qualitative results on all six captures. Our evaluation focuses on (i) the overfitting capabilities of each method on the training data, (ii) the ability of our encoder to generalize to unseen textures, (iii) temporal stability and visual plausibility of our appearance predictor compared to the baselines w.r.t. spurious correlations and pose–appearance ambiguity. We strongly encourage the reader to watch the supplementary videos, as the improvements in temporal stability and motion dynamics plausibility are more easily observable in videos than in still images or aggregate metrics.

Quantitative. We report standard reconstruction metrics, including PSNR, SSIM [38], and the perceptual similarity metric LPIPS [47]. Tab. 1 summarizes training- and test-set performance for all compared methods. On the training set, our method achieves the best scores, indicating that the added structure (localized conditioning and appearance latents) enables high-quality fitting of the

Table 2: Ablation of our components.

	Test			Train		
	PSNR$^{\uparrow}$	SSIM$^{\uparrow}$	LPIPS$^{\downarrow}$	PSNR$^{\uparrow}$	SSIM$^{\uparrow}$	LPIPS$^{\downarrow}$
Ours	32.99	0.939	0.063	36.21	0.950	0.053
Ours w/o local pca	32.86	0.938	0.064	36.23	0.950	0.053
Ours w/o local pose	32.87	0.938	0.064	36.29	0.951	0.052

training data. On the test set, our full pipeline with the appearance predictor performs best overall. Because appearance is inherently ambiguous and drifts over time, we expect the appearance immediately after initialization to match the ground truth closest. Accordingly, we report results with the predictor re-initialized every 30 frames, which shows larger improvements over this interval. Additionally, we report a test-time upper bound using the appearance encoder, which isolates the encoder and highlights its ability to generalize to novel poses and previously unseen textures. We also find that temporal stability under driving improves substantially in practice, as demonstrated in the qualitative results and supplementary videos. Finally, we further ablate our design choices in Tab. 2. Applying localized pose masking and localized PCA slightly reduces training-set fit, but improves generalization when driving with novel poses.



(a) MMLPs† **(b)** Ours

Fig. 6: Qualitative comparison of **global** (MMLPs†) versus **localized** (Ours) pose conditioning. We manually perturb a single pose parameter (top: neck twist; bottom: wrist rotate) between two frames: global conditioning induces large, non-local deformations and spurious appearance changes (visualized as ℓ_1 error in the cutouts), whereas localized conditioning confines the effect to local, physically plausible motion.

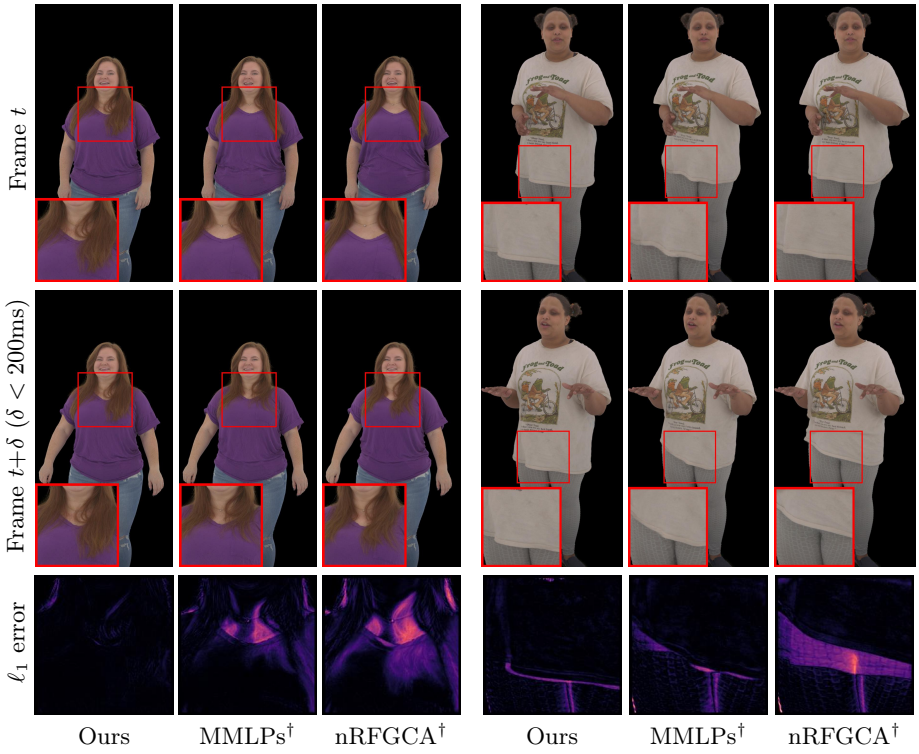


Fig. 7: Two temporally adjacent frames from a test sequence and the corresponding cutout ℓ_1 error against the previous frame. Our method remains temporally stable due to explicit pose–appearance disentanglement and the learned appearance predictor, while nRFGCA† and MMLPs† exhibit abrupt, implausible appearance changes.

Qualitative. Fig. 6 illustrates how our localized pose conditioning reduces spurious long-range correlations. *MMLPs*† conditions every anchor on the full pose vector, allowing the decoder to exploit incidental correlations in the training data; as a result, distant pose changes can spuriously modulate unrelated appearance, such as arm shadows, shirt wrinkles, or hair configuration. In contrast, our approach supplies pose parameters only locally and applies localized PCA consistently at both training and test time, which suppresses unrealistic long-range effects. Across captures, we observe that *MMLPs*† is particularly prone to unstable shading and deformation due to the test-time-only global PCA, while access to global pose parameters further encourages spurious correlations.

To assess temporal stability, Fig. 7 shows two temporally adjacent frames from a test sequence. Both *nRFGCA*† and *MMLPs*† can exhibit noticeable flicker on challenging captures, with abrupt and implausible appearance changes over very short time spans. Our method, driven by the appearance predictor, produces substantially more stable results while still allowing realistic, pose-dependent

local appearance evolution from a short motion history. This effect is most clearly visible in the supplementary videos.

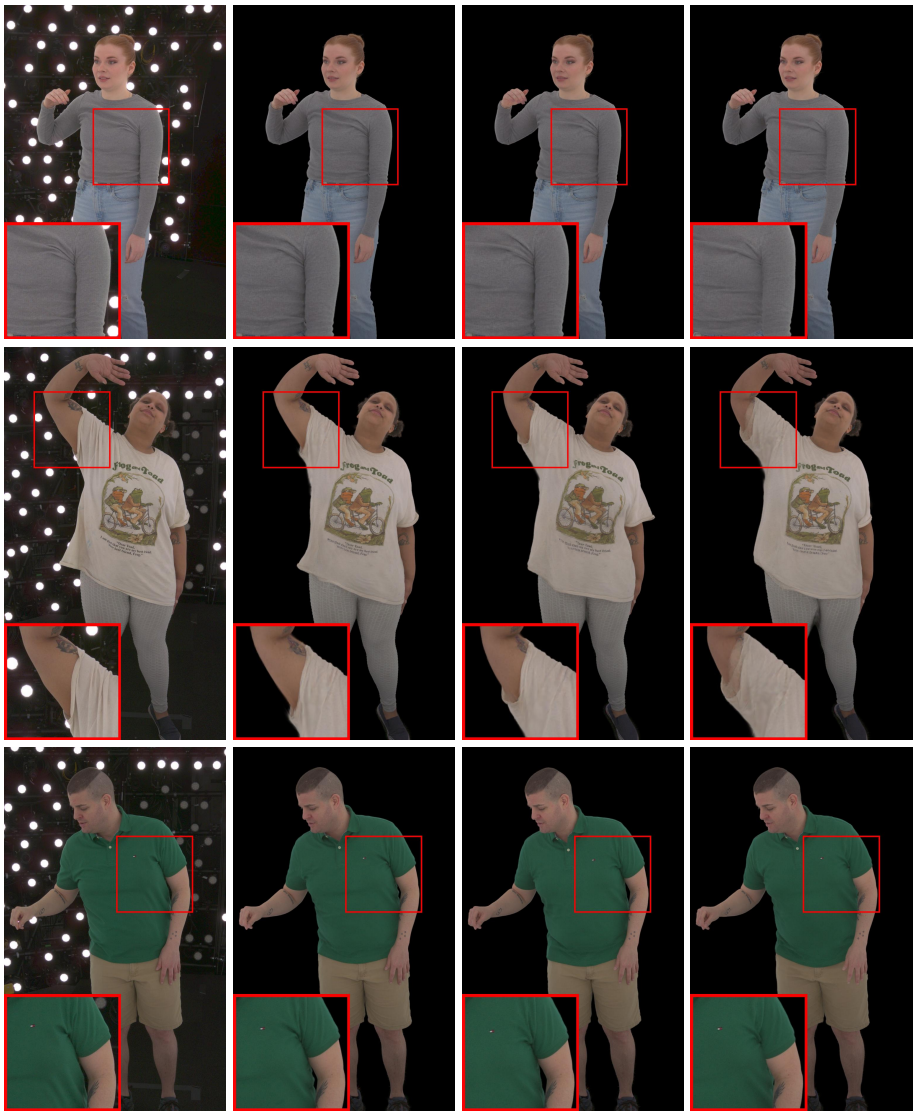


Fig. 8: Test-sequence comparison of our method using the texture encoder (unseen texture input) versus our appearance predictor (no textures), alongside MMLPs†. The encoder provides an upper bound closest to the ground truth, while the predictor yields temporally smooth, plausible appearance evolution without texture inputs.

Finally, Fig. 8 highlights (i) the generalization of our appearance encoder and (ii) the behavior of the appearance predictor on unseen sequences. When provided with projected UV textures, the encoder produces latents that yield renderings closest to the ground truth, despite never observing these exact textures during training. The predictor generates temporally smooth and plausible appearance changes without texture input; while it does not always match the ground-truth appearance (due to the inherent one-to-many ambiguity), it avoids the erratic switching and shading artifacts commonly observed in $MMLPs^{\dagger}$.

5 Limitations & Future Work

While our appearance predictor yields stable and plausible results in most cases, several limitations remain. First, despite explicitly introducing appearance latents, our current formulation does not fully disentangle pose-driven effects from appearance variation. In practice, the predictor may absorb factors that are, in principle, deterministically explainable from pose. Our efforts of enforcing a stricter separation in our experiments—similar to [1]—consistently reduced reconstruction quality and limited the model's ability to fit fine details.

Second, strictly localized pose conditioning cannot represent certain long-range interactions, such as cast shadows of the arms onto the torso. As a result, these effects can be encoded in the appearance latents, which may reduce robustness under extreme out-of-distribution motion. A promising direction is modeling long-range shading separately—*e.g.*, via an explicit, low-frequency lighting or shadow component [37]—thereby reducing the burden on the appearance codes and improving stability when extrapolating beyond the training distribution.

Finally, the initialization of the appearance predictor is currently simplistic: we either use encoder-produced codes when a UV texture is available, or fall back to a fixed zero code otherwise. Although the predictor is trained to recover from this setting over the first few frames, it introduces a short "ramp-up" period before reaching a coherent appearance. While this can be hidden from the user, a more principled generative initialization—*e.g.*, conditioned on the first pose(s)—could yield realistic initial appearances without requiring this warm-up.

Future work should therefore focus on stronger pose–appearance disentanglement, explicit modeling of long-range shading effects, and improved test-time initialization for appearance prediction.

6 Conclusion

We presented a 3D Gaussian Splatting avatar model that improves both reconstruction fidelity and test-time stability by decoupling appearance variation from pose. By learning per-frame appearance latents from reconstructed UV textures, our method builds a compact and semantically meaningful appearance space that captures time-varying details such as cloth, wrinkles, and hair. We further showed that an autoregressive predictor operating in this latent space enables temporally coherent and visually plausible appearance evolution during

driving without texture inputs. Finally, our principled localization of pose parameters reduces spurious pose–appearance correlations, improving robustness under novel motions and yielding more stable downstream animation.

References

1. Bagautdinov, T., Wu, C., Simon, T., Prada, F., Shiratori, T., Wei, S.E., Xu, W., Sheikh, Y., Saragih, J.: Driving-signal aware full-body avatars. ACM TOG **40**(4) (2021) 3, 14
2. Casas, D., Volino, M., Collomosse, J., Hilton, A.: 4D Video Textures for Interactive Character Appearance. Comput. Graph. Forum **33**(2), 371–380 (2014) 3
3. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: TensoRF: Tensorial Radiance Fields. In: ECCV. pp. 333–350 (2022) 4
4. Chen, J., Hu, J., Wang, G., Jiang, Z., Zhou, T., Chen, Z., Lv, C.: TaoAvatar: Real-Time Lifelike Full-Body Talking Avatars for Augmented Reality via 3D Gaussian Splatting. In: CVPR. pp. 10723–10734 (2025) 4
5. Chen, Y., Zheng, Z., Li, Z., Xu, C., Liu, Y.: MeshAvatar: Learning High-Quality Triangular Human Avatars from Multi-view Videos. In: ECCV (2024) 4
6. Deng, X., Zheng, Z., Zhang, Y., Sun, J., Xu, C., Yang, X., Wang, L., Liu, Y.: RAM-Avatar: Real-time Photo-Realistic Avatar from Monocular Videos with Full-body Control. In: CVPR. pp. 1996–2007 (2024) 3
7. Ferguson, A., Osman, A.A.A., Bescos, B., Stoll, C., Twigg, C., Lassner, C., Otte, D., Vignola, E., Prada, F., Bogo, F., Santesteban, I., Romero, J., Zarate, J., Lee, J., Park, J., Yang, J., Doublestein, J., Venkateshan, K., Kitani, K., Kavan, L., Farra, M.D., Hu, M., Cioffi, M., Fabris, M., Ranieri, M., Modarres, M., Kadlecek, P., Khirodkar, R., Abdrashitov, R., Prévost, R., Rajbhandari, R., Mallet, R., Pearsall, R., Kao, S., Kumar, S., Parrish, S., Yu, S.I., Saito, S., Shiratori, T., Wang, T.L., Tung, T., Xu, Y., Dong, Y., Chen, Y., Xu, Y., Ye, Y., Jiang, Z.: Mhr: Momentum human rig (2025), https://arxiv.org/abs/2511.15586 5
8. Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., Black, M.J.: DRAPE: DRessing Any PErson. ACM TOG **31**(4) (2012) 3
9. Habermann, M., Liu, L., Xu, W., Zollhoefer, M., Pons-Moll, G., Theobalt, C.: Real-time deep dynamic characters. ACM TOG **40**(4) (2021) 3
10. Habermann, M., Xu, W., Zollhoefer, M., Pons-Moll, G., Theobalt, C.: LiveCap: Real-Time Human Performance Capture From Monocular Video. ACM TOG **38**(2) (2019) 3
11. Hu, L., Zhang, H., Zhang, Y., Zhou, B., Liu, B., Zhang, S., Nie, L.: GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians. In: CVPR. pp. 634–644 (2024) 3
12. Hu, S., Hu, T., Liu, Z.: GauHuman: Articulated Gaussian Splatting from Monocular Human Videos. In: CVPR. pp. 20418–20431 (2024) 3
13. Iandola, F., Pidhorskyi, S., Santesteban, I., Gupta, D., Pahuja, A., Bartolovic, N., Yu, F., Garbin, E., Simon, T., Saito, S.: Squeezeme: Mobile-ready distillation of gaussian full-body avatars. In: ACM SIGGRAPH Conference Papers (2025) 4
14. Işık, M., Rünz, M., Georgopoulos, M., Khakhulin, T., Starck, J., Agapito, L., Nießner, M.: HumanRF: High-Fidelity Neural Radiance Fields for Humans in Motion. ACM TOG **42**(4) (2023) 3
15. Jiang, T., Chen, X., Song, J., Hilliges, O.: InstantAvatar: Learning Avatars From Monocular Video in 60 Seconds. In: CVPR. pp. 16922–16932 (2023) 3
16. Jiang, W., Yi, K.M., Samei, G., Tuzel, O., Ranjan, A.: NeuMan: Neural Human Radiance Field from a Single Video. In: ECCV. pp. 402–418 (2022) 3
17. Jiang, Y., Liao, Q., Li, X., Ma, L., Zhang, Q., Zhang, C., Lu, Z., Shan, Y.: UV Gaussians: Joint Learning of Mesh Deformation and Gaussian Textures for Human Avatar Modeling. Knowledge-Based Systems **320** (2025) 4

18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM TOG **42**(4) (2023) 1, 4
19. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114 (2013) 21
20. Kocabas, M., Chang, J.H.R., Gabriel, J., Tuzel, O., Ranjan, A.: HUGS: Human Gaussian Splats. In: CVPR. pp. 505–515 (2024) 4
21. Lei, J., Wang, Y., Pavlakos, G., Liu, L., Daniilidis, K.: GART: Gaussian Articulated Template Models. In: CVPR. pp. 19876–19887 (2024) 4
22. Li, R., Tanke, J., Vo, M., Zollhöfer, M., Gall, J., Kanazawa, A., Lassner, C.: TAVA: Template-free Animatable Volumetric Actors. In: ECCV. pp. 419–436 (2022) 4
23. Li, Z., Zheng, Z., Wang, L., Liu, Y.: Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling. In: CVPR (2024) 4, 5, 7
24. Lin, S., Zhang, H., Zheng, Z., Shao, R., Liu, Y.: Learning Implicit Templates for Point-Based Clothed Human Modeling. In: ECCV. pp. 210–228 (2022) 4
25. Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: neural free-view synthesis of human actors with pose control. ACM TOG **40**(6) (2021) 3
26. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. In: International Conference on 3D Vision (3DV). pp. 800–809 (2024) 3
27. Ma, Q., Yang, J., Tang, S., Black, M.J.: The Power of Points for Modeling Humans in Clothing. In: ICCV. pp. 10974–10984 (2021) 4
28. Ma, S., Simon, T., Saragih, J., Wang, D., Li, Y., De La Torre, F., Sheikh, Y.: Pixel Codec Avatars. In: CVPR (2021) 3
29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: ECCV (2020) 3
30. Moreau, A., Song, J., Dhamo, H., Shaw, R., Zhou, Y., Pérez-Pellitero, E.: Human Gaussian Splatting: Real-time Rendering of Animatable Avatars . In: CVPR. pp. 788–798 (2024) 4
31. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG **41**(4) (2022) 4
32. Pang, H., Zhu, H., Kortylewski, A., Theobalt, C., Habermann, M.: ASH: Animatable Gaussian Splats for Efficient and Photoreal Human Rendering. In: CVPR. pp. 1165–1175 (2024) 4
33. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable Neural Radiance Fields for Modeling Dynamic Human Bodies. In: ICCV. pp. 14314–14323 (2021) 3
34. Peng, S., Geng, C., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Zhou, X., Bao, H.: Implicit Neural Representations With Structured Latent Codes for Human Body Modeling. IEEE TPAMI **45**(8), 9895–9907 (2023) 3
35. Qian, Z., Wang, S., Mihajlovic, M., Geiger, A., Tang, S.: 3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting. In: CVPR. pp. 5020–5030 (2024) 4
36. Shao, Z., Wang, Z., Li, Z., Wang, D., Lin, X., Zhang, Y., Fan, M., Wang, Z.: SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In: CVPR. pp. 1606–1616 (2024) 4
37. Wang, S., Simon, T., Santesteban, I., Bagautdinov, T., Li, J., Agrawal, V., Prada, F., Yu, S.I., Nalbone, P., Gramlich, M., et al.: Relightable Full-body Gaussian Codec Avatars. In: ACM SIGGRAPH Conference Papers (2025) 4, 9, 14, 23

38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004) 5, 10
39. Weng, C.Y., Curless, B., Srinivasan, P.P., Barron, J.T., Kemelmacher-Shlizerman, I.: HumanNeRF: Free-Viewpoint Rendering of Moving People From Monocular Video. In: CVPR. pp. 16210–16220 (2022) 3
40. Xiang, D., Bagautdinov, T., Stuyck, T., Prada, F., Romero, J., Xu, W., Saito, S., Guo, J., Smith, B., Shiratori, T., et al.: Dressing Avatars: Deep Photorealistic Appearance for Physically Simulated Clothing. ACM TOG **41**(6) (2022) 3
41. Xiang, D., Prada, F., Bagautdinov, T., Xu, W., Dong, Y., Wen, H., Hodgins, J., Wu, C.: Modeling clothing as a separate layer for an animatable human avatar. ACM TOG **40**(6) (2021) 3
42. Xiang, D., Prada, F., Cao, Z., Guo, K., Wu, C., Hodgins, J., Bagautdinov, T.: Drivable Avatar Clothing: Faithful Full-Body Telepresence with Dynamic Clothing Driven by Sparse RGB-D Input. In: ACM SIGGRAPH Asia 2023 Conference Papers (2023) 3
43. Xu, F., Liu, Y., Stoll, C., Tompkin, J., Bharaj, G., Dai, Q., Seidel, H.P., Kautz, J., Theobalt, C.: Video-based characters: creating new human performances from a multi-view video database. In: ACM SIGGRAPH Conference Papers (2011) 3
44. Xu, H., Alldieck, T., Sminchisescu, C.: H-NeRF: Neural Radiance Fields for Rendering and Temporal Reconstruction of Humans in Motion. In: NeurIPS. vol. 34, pp. 14955–14966 (2021) 3
45. Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., Kanazawa, A.: gsplat: An open-source library for Gaussian splatting. Journal of Machine Learning Research **26**(34) (2025) 9
46. Zhan, Y., Shao, T., Yang, Y., Zhou, K.: Real-time High-fidelity Gaussian Human Avatars with Position-based Interpolation of Spatially Distributed MLPs. In: CVPR (2025) 2, 4, 5, 7, 9, 10, 19, 21, 22, 23
47. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: CVPR (2018) 10
48. Zhang, X., Zhang, J., Chacko, R., Xu, H., Song, G., Yang, Y., Feng, J.: GETAvatar: Generative Textured Meshes for Animatable Human Avatars. In: ICCV. pp. 2273–2282 (2023) 3
49. Zheng, Y., Zhao, Q., Yang, G., Yifan, W., Xiang, D., Dubost, F., Lagun, D., Beeler, T., Tombari, F., Guibas, L., et al.: PhysAvatar: Learning the Physics of Dressed 3D Avatars from Visual Observations. In: ECCV. pp. 262–284 (2024) 3
50. Zheng, Z., Zhao, X., Zhang, H., Liu, B., Liu, Y.: AvatarReX: Real-time Expressive Full-body Avatars. ACM TOG **42**(4) (2023) 3
51. Zhu, H., Qiu, L., Qiu, Y., Han, X.: Registering Explicit to Implicit: Towards High-Fidelity Garment mesh Reconstruction from Single Images. In: CVPR. pp. 3845–3854 (2022) 3
52. Zielonka, W., Bagautdinov, T., Saito, S., Zollhöfer, M., Thies, J., Romero, J.: Drivable 3D Gaussian Avatars. In: International Conference on 3D Vision (3DV). pp. 979–990 (2025) 4

Supplementary Material

Autoregressive Appearance Prediction for 3D Gaussian Avatars

A Spatial MLPs Preliminaries

Zhan et al. [46] construct a hierarchical structure of points by sampling a number of anchors/control points/Gaussians (300/10k/200k) on the template mesh. During initialization, they calculate weights $t_{i,j}$ for the i-th Gaussian at position $\mathbf{x}_i$ to the three closest anchors in their neighborhood $\mathcal{N}_3$ based on the reciprocal distance $d_{i,j}$ to each anchor at position $\mathbf{x}_j^a$:

$$t_{i,j} = \frac{d_{i,j}}{\sum_{k \in \mathcal{N}_3} d_{i,k}}, \quad \text{with} \quad d_{i,j} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j^a\|_2}. \tag{3}$$

Each anchor holds an MLP mapping pose parameters $\boldsymbol{\theta}$ to corrective weights $\mathbf{w}^a \in \mathbb{R}^B$. Every Gaussian then computes its corrective weights $\mathbf{w}_i$ as the weighted average $\mathbf{w}_i = \sum_{j \in \mathcal{N}_3} t_{i,j} \mathbf{w}_j^a$. The corrective weights of the control points $\mathbf{w}^c$ are computed using the same procedure. Additionally, each Gaussian holds offset vectors $\boldsymbol{\delta\Lambda_k} \in \mathbb{R}^B$ and a bias λ_k for each of the Gaussian properties' values $\{\mathbf{r}, \mathbf{s}, \mathbf{c}, o\}$. The actual values of each property p_k (e.g. scale in x-dimension $\mathbf{s_x}$) of the i-th Gaussian in canonical space are then computed as

$$p_k = h_k(\lambda_k + \delta\lambda_k), \quad \text{with} \quad \delta\lambda_k = \langle \mathbf{w}_i, \boldsymbol{\delta\Lambda_k} \rangle, \tag{4}$$

with h_k being a property-specific activation function. Equivalently, each control point stores bias terms to compute its positional offset $\boldsymbol{\delta x^c}$. Each Gaussian then computes its positional offset $\boldsymbol{\delta x}$ from the weighted average of the three closest control points $\mathcal{N}_3^c$ (similar to Eq. (3) and $\mathbf{w}_i$) plus a small per-Gaussian offset $\boldsymbol{\delta x}_0$. Adding this offset to the initial position of the Gaussian on the template mesh gives its final position in canonical space. Finally, the canonical Gaussians are then posed using linear blend skinning (LBS).

B Dataset Details

B.1 Capture Info

Fig. 9 presents the six captures evaluated in our experiments. The captures span a range of difficulty factors, including long hair (*Actor 2 and 5*), loose clothing (*Actor 3 and 5*), untucked shirts (*Actor 1 and 4*), and tight yet wrinkled garments (*Actor 6*). Each actor is recorded for ∼30 minutes, from which we select 12–20 minutes of segments exhibiting high pose diversity. The captures further differ in frame rate, ranging from 24–45 FPS, which results in 20–35k training frames. We evaluate the appearance predictor on the test sequences at the training frame rate. Running the appearance predictor at a frame rate different from the one used during training may require additional measures.

(a) Actor 1 **(b)** Actor 2 **(c)** Actor 3

(d) Actor 4 **(e)** Actor 5 **(f)** Actor 6

Fig. 9: Our dataset consists of six captures of varying difficulty.

B.2 UV Texture Projection

We use the mean shape of a subject as the template mesh and pose it using per-frame tracked body poses. For each view, we rasterize the posed mesh and project the image pixels to the mesh's texels. To minimize the impact of inaccurate surface boundaries, we apply eroded segmentation masks to identify non-boundary pixels and only project these pixels. When fusing texel intensities from multiple views, we first compute the median and filter out values outside the range $[0.8, 1.1]$ times the median intensity. The final texel value is then calculated by averaging the remaining intensities. Through experimentation, we find that this approach achieves reasonable spatial and temporal consistency.

C Training Details

C.1 Gaussian Model

Following Zhan et al. [46] we initialize 300 anchors, 10k control points, and 200 Gaussians on the template mesh. Each anchor is associated with an MLP with hidden layer sizes $\{512, 256, 256, 256\}$, which outputs 16 coefficients for the Gaussian corrective basis and 16 coefficients for the control point basis. We apply nonlinearities to each Gaussian property: an exponential activation for scale, a sigmoid for opacity, and a tanh for spherical harmonics (scaled such that sh0 can represent RGB values in the range $[0, 1]$). We do not apply correctives to opacity, and we additionally max-clamp scale at 0.2 to encourage a more coherent and stable representation. Training is performed for 100k iterations; we increase the spherical-harmonics degree from 0 to 1 after 60k iterations and enable correctives after 500 iterations.

We use the following learning rates for the (bias term / corrective basis) of each property: scale ($5\times10^{-4}/1\times10^{-4}$), rotation ($5\times10^{-4}/1\times10^{-4}$), opacity ($5\times10^{-4}/-$), sh0 ($2.5\times10^{-3}/1\times10^{-4}$), shN ($2.5\times10^{-5}/2.5\times10^{-6}$), control point position ($1.6\times10^{-4}, 1.6\times10^{-5}$), and Gaussian position ($-/1\times10^{-4}$). The anchor MLP is trained with a learning rate of 5×10^{-4} and a weight decay of 1×10^{-3}.

C.2 Appearance Encoder

The encoder takes a $1024\times1024\times3$ RGB image as input and produces a $32\times32\times N_l$ feature map, where the size of the latent codes $N_l = 16$. We use a CNN composed of multiple down-convolution layers with kernel size $k = 3$ and stride 2, followed by 8-wide group normalization and ReLU activations. The feature dimension is doubled at each layer (except for the first layer, which increases the dimension from 3 to 8), until reaching a maximum of 128 channels and a spatial resolution of 32×32. After downsampling, two separate 3×3 convolutional layers predict N_l-dimensional maps for μ and $\log(\sigma^2)$, from which we sample per-texel spatial latent codes from a normal distribution using the re-parametrization trick [19]. The encoder is trained with a learning rate of 5×10^{-4} and weight decay 1×10^{-3}.

C.3 Appearance Predictor

The appearance predictor transformer uses a single transformer block with 4 attention heads, token (embedding) dimension 128, and feed-forward hidden dimension 256. Each pose token is produced by a dedicated shallow MLP, instantiated separately for each pose parameter and modality, while each appearance token is produced by a dedicated shallow MLP per anchor. We apply sinusoidal absolute positional encoding in the transformer block. The model output is generated by a linear projection head that maps the 128-dimensional token representations back to 16-dimensional per-anchor latent codes. All components of the appearance predictor are trained using a learning rate of 1×10^{-3}.

D Additional Qualitative Evaluation

We evaluate the generalization capabilities of our encoder to unseen textures in Fig. 10. Our encoder is able to deliver results that are close to the ground truth, even though it might not have seen the exact hair configuration or wrinkle pattern during training.

Fig. 10: Multiple image pairs showing the generalization capabilities of our texture encoder on the test set, *i.e.*, novel poses and unseen UV textures. The left image shows ground truth and the right image is the appearance encoder result.

We show additional results in Fig. 11 for our encoder and the appearance predictor on a test sequence, compared to the baseline *MMLPs*† [46]. While the results from our appearance encoder are closer to the ground truth, the predictor results provide a believable appearance evolution and are more temporally stable than *MMLPs*†, which suffers from spurious correlations and pose–appearance ambiguities.

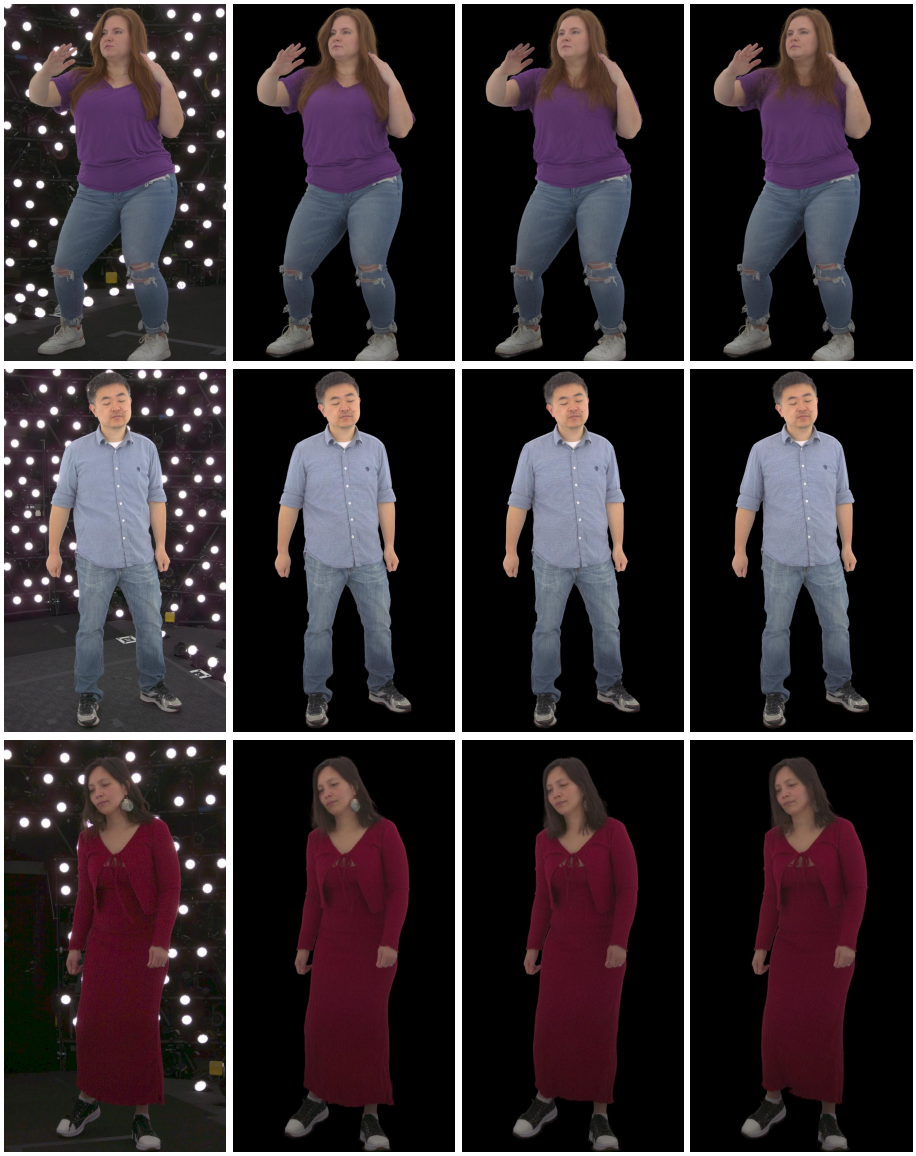


Fig. 11: Test-sequence comparison of our method using the texture encoder (unseen texture input) versus our appearance predictor (no textures), alongside MMLPs†.

E Per-Capture Quantitative Results

We show the per-capture quantitative results on the test and training set against *MMLPs*† [46] and *nRFGCA*† [37] in Tab. 3 and Tab. 4.

Table 3: Test set image metrics on all compared methods. We initialize the appearance predictor for Ours with the encoder output and show additional results for re-initializing every 30 frames, and for using the texture encoder on the test sequences.

PSNR$^{\uparrow}$	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Mean
Ours	31.52	31.90	28.82	35.03	34.38	36.31	32.99
MMLPs†	31.29	31.79	28.41	33.68	34.31	34.55	32.34
nRFGCA†	31.47	31.94	28.95	34.84	34.29	34.97	32.74
Ours (re-init. 30)	32.39	32.93	30.04	35.80	35.02	36.92	33.85
Ours (encoder)	33.33	33.85	31.48	36.71	35.63	37.67	34.78

SSIM$^{\uparrow}$	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Mean
Ours	0.923	0.942	0.928	0.950	0.931	0.959	0.939
MMLPs†	0.922	0.941	0.927	0.947	0.930	0.953	0.937
nRFGCA†	0.924	0.942	0.929	0.950	0.932	0.956	0.939
Ours (re-init. 30)	0.927	0.946	0.934	0.952	0.932	0.962	0.942
Ours (encoder)	0.933	0.950	0.942	0.954	0.934	0.965	0.946

LPIPS$^{\downarrow}$	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Mean
Ours	0.062	0.064	0.082	0.066	0.061	0.045	0.063
MMLPs†	0.063	0.066	0.084	0.071	0.062	0.051	0.066
nRFGCA†	0.063	0.065	0.083	0.070	0.059	0.050	0.065
Ours (re-init. 30)	0.059	0.060	0.075	0.063	0.059	0.043	0.060
Ours (encoder)	0.055	0.056	0.068	0.060	0.057	0.040	0.056

Table 4: Training set image metrics on all compared methods.

PSNR$^{\uparrow}$	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Mean
Ours	34.97	35.40	34.58	37.14	36.52	38.63	36.21
MMLPs†	35.00	35.36	34.22	37.07	36.51	38.62	36.13
nRFGCA†	34.69	35.58	33.42	36.84	36.53	38.56	35.94

SSIM$^{\uparrow}$	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Mean
Ours	0.939	0.956	0.949	0.956	0.934	0.969	0.950
MMLPs†	0.939	0.956	0.947	0.955	0.934	0.969	0.950
nRFGCA†	0.939	0.957	0.943	0.955	0.935	0.969	0.950

LPIPS$^{\downarrow}$	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Mean
Ours	0.052	0.053	0.061	0.058	0.057	0.038	0.053
MMLPs†	0.053	0.053	0.064	0.059	0.057	0.038	0.054
nRFGCA†	0.054	0.052	0.067	0.062	0.054	0.038	0.055