RegHead: Non-Humanoid Head Blendshapes via Feed-Forward Registration

Jiahao Luo[1], Hao Zhang[2], Jianqi Chen[3], Yijie He[4], Jiaxu Zou[4], Michael Vasilkovsky[4], Sergei Korolev[4], Sergey Tulyakov[4], Chaoyang Wang[4], Peter Wonka[3,4], James Davis[1], and Jian Wang[4]

[1]UCSC [2]UIUC [3]KAUST [4]Snap Inc.

Abstract. We present RegHead, a framework for constructing semantic blendshape sets for animatable non-humanoid head avatars. With a fixed expression vocabulary, semantic blendshapes provide a low-dimensional and interpretable animation interface and support cross-identity retargeting. Building such blendshape sets remains expensive because (i) expression-consistent supervision is scarce, (ii) generated 4D assets typically lack correspondence, and (iii) facial motion is highly localized. We propose (1) a large-scale dataset of non-humanoid identities paired with a shared expression vocabulary, obtained by expanding a small artist-rigged library via fine-tuned image editing; (2) a dense stochastic anchor motion representation tailored to localized facial deformations; and (3) a fast feed-forward registration model that converts unregistered expression meshes into a corresponded blendshape basis by predicting anchor-based deformations from the neutral shape. Experiments show that our approach produces higher-fidelity expression meshes than baselines, while running orders of magnitude faster than optimization. We further demonstrate real-time retargeting from human face tracking signals to non-humanoid characters, capturing both head pose and localized facial motions. Our project page is available at `https://snap-research.github.io/RegHead/`.

1 Introduction

The creation of animatable 3D non-humanoid head avatars is increasingly important for AR communication [13, 38], social media [2, 26], and games [10, 11], yet remains underexplored. A practical animation interface for non-humanoid heads should support controllable expressions and efficient retargeting across diverse identities, species, and stylized appearances. *Semantic blendshapes* provide such an interface by defining a fixed *expression vocabulary*, a small set of interpretable expression representations in full correspondence [14, 15, 23]. Given this vocabulary, animation reduces to predicting a low-dimensional weight vector over expressions and linearly blending the corresponding blendshapes, which naturally supports real-time control. Moreover, because the same vocabulary is shared across identities, expression weights carry consistent semantic meaning, enabling cross-identity retargeting without solving a new correspondence

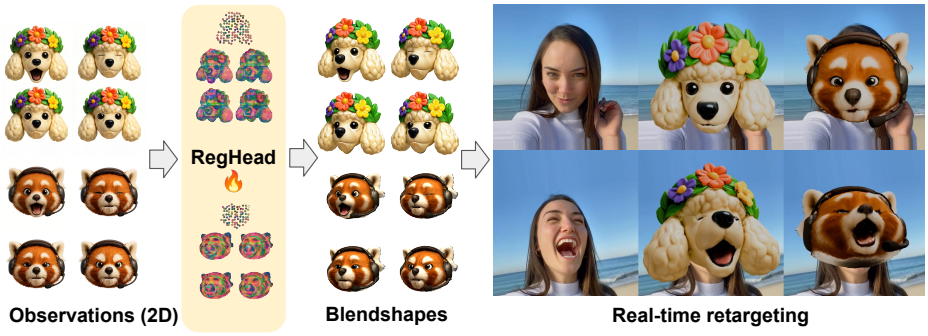

Fig. 1: RegHead converts semantically labeled expression observations into a corresponded semantic blendshape set for non-humanoid heads in a single feed-forward pass. The resulting blendshapes support real-time animation and retargeting via a fixed expression vocabulary.

problem at animation time. Despite these advantages, constructing semantic blendshape sets for diverse non-humanoid heads requires substantial artist effort [14, 15] or heavy optimization-based processes [4, 22, 30].

Building semantic non-humanoid blendshapes in a feed-forward manner is challenging due to several underexplored bottlenecks. 1) Current image/video generation and editing models [1, 20, 39, 46] do not reliably produce the *same* predefined expression (e.g., a specific "frown" or "eye-closed") across different identities, making it hard to obtain large-scale 2D observations with consistent semantic labels. 2) Modern 3D/4D generation methods [29, 36, 41, 44, 45, 47] can produce high-quality expression-specific surfaces, but their outputs typically lack vertex correspondence across expressions, preventing direct use as a blendshape basis. Recovering correspondence is possible with optimization-heavy non-rigid registration or per-instance shape matching, but these approaches are expensive and difficult to scale. 3) Non-humanoid facial expressions often involve highly localized, species-dependent deformations (e.g., jaw articulation, eyelids) that are poorly captured by generic global deformation models [19, 31, 51, 52]. As a result, the core challenge is not only generating expression-specific assets, but recovering *semantically consistent correspondence* while preserving fine local deformations in a scalable pipeline.

RegHead addresses these challenges with three components: a large-scale non-humanoid head dataset (Sec. 3.1), a dense stochastic-anchor motion representation (Sec. 3.2), and a feed-forward registration model (Sec. 3.3). We begin by defining a semantic expression vocabulary $\mathcal{E}$ using a small set of high-quality, artist-rigged non-humanoid head assets, which provides consistent semantic observations for blendshape construction. We then scale this vocabulary to many identities by fine-tuning an image editing model [39]. The resulting expression-specific meshes $\{\tilde{\mathcal{M}}^t\}_{t=0}^{T}$ are visually plausible but not in vertex correspondence.

The key novelty of RegHead is to amortize correspondence recovery into a learned feed-forward registration model, instead of solving a new per-instance

optimization problem. RegHead predicts the deformation from the neutral mesh to each target expression in a single pass. To handle diverse non-humanoid facial topology and anatomy, we avoid predefined skeletons, bones, or templates, and introduce dense stochastic anchors as a topology-agnostic motion representation. To make the prediction reliable for both large-support motions and fine localized facial changes, RegHead combines a global matcher for coarse alignment with a structured local matcher for fine-grained correspondence refinement. The model is trained without ground-truth point correspondences or deformation fields, using rendered target consistency as supervision.

Experiments show that RegHead produces higher-fidelity expression meshes than feed-forward and optimization-based baselines, while running orders of magnitude faster than optimization methods. The resulting semantic blendshapes further enable real-time retargeting from human face tracking to non-humanoid characters.

The main contributions of RegHead are:

- We construct a large-scale dataset of ∼20k non-humanoid identities with a fixed semantic expression vocabulary, enabling scalable study of non-humanoid head blendshape generation.
- We introduce dense stochastic anchors, a topology-agnostic motion representation for localized non-humanoid facial deformation without predefined bones, skeletons, or shared templates.
- We propose a feed-forward registration model that converts unregistered expression meshes into corresponded semantic blendshapes in a single pass, trained without ground-truth correspondences or deformation fields.
- We demonstrate improved fidelity and speed over feed-forward and optimization-based baselines, and enable real-time retargeting from human face tracking to non-humanoid characters.

2 Related Works

2.1 Head Avatars

Recent advancements have extended human head avatar pipelines [5,7,21,25,28, 43] to stylized or toonified characters through text or style conditioning [27, 32, 35,37,48]. Among these, TextToon [35] generates a drivable toonified head avatar from a monocular video and text style. LeGO [48] generates a stylized 3D face model with a surface deformation network on a 3D morphable model (3DMM). HeadEvolver [37] generates stylized head avatars via template mesh deformation and 2D diffusion priors. However, these methods typically require parametric priors such as 3DMM. While effective for humanoids, these methods struggle when the target geometry deviates significantly from human anatomy. T2Bs [22] animates character heads by aligning static assets to diffusion-generated videos via deformable 3DGS. While this bypasses some template constraints, it depends on a video-generation-and-alignment pipeline and optimization-heavy deformation baking. In contrast, we propose a fully feed-forward framework that avoids template-based anatomical constraints while achieving fast inference speed.

2.2 4D Asset Reconstruction

Traditional 4D asset reconstruction methods [8, 49] primarily rely on Score Distillation Sampling (SDS) with priors from pretrained image and video diffusion models. Although subsequent works [18, 29, 44] have made significant progress in bypassing SDS, they mainly reconstruct 4D content using implicit representations such as NeRF [24] or 3DGS [12]. These approaches generally don't explicitly enforce temporal correspondence across frames, which limits their applicability in industrial scenarios, including AR and gaming, where topology-consistent assets are often required. To address temporal correspondence issues, methods such as DreamMesh4D [17] and V2M4 [4] adopt mesh-based representations and recover topology-consistent geometry by coupling reconstruction with per-sequence alignment and refinement. While they achieve correspondence consistency, they rely on optimization-based pipelines, leading to substantial test-time computation. To improve efficiency, more recent works [3, 9, 31, 33] propose feed-forward frameworks that maintain correspondence by predicting deformation fields over an anchor mesh. DriveAnyMesh [33] extends 3DShape2VecSet [50] to condition on temporally consistent multi-view frames and predicts deformations of anchor point clouds. ActionMesh [31] introduces a two-stage network that first reconstructs structure-aligned meshes and then predicts deformations relative to an anchor mesh. These approaches achieve strong performance in both speed and quality. However, they primarily focus on articulated or whole-object motion and do not explicitly address the highly localized, fine-grained, and species-dependent deformations required for non-humanoid head animation. In contrast, our method is specifically designed to model such fine-grained deformations. We introduce a tailored motion representation along with global and local matching modules to effectively capture fine-grained non-humanoid facial movements while maintaining fast inference speed.

2.3 Animation Representation

In parallel with direct vertex-level deformation methods [4, 31, 33], another line of research focuses on preparing animation-ready assets by predicting skeletons and skinning weights, enabling mesh animation through skeletal transformations. Methods such as RigAnything [19], MagicArticulate [34], and UniRig [52] automatically generate skeleton structures and corresponding skinning weights for diverse 3D assets, making static meshes articulation-ready at scale. RigMo [51] further extends this direction by jointly predicting rig structures and motion from mesh sequences, facilitating generic character animation directly from data. However, the predicted skeletons in these methods are typically sparse and designed for coarse articulated motion. As a result, they are not well suited for non-humanoid head animation, which requires highly localized and fine-grained deformations. In contrast, our method adopts a motion representation based on a dense set of anchor points to drive mesh deformation, enabling more precise modeling of subtle and detailed movements.

3 Methods

Our goal is to generate a semantically defined blendshape set $\{\mathcal{B}^t\}_{t=0}^T$ associated with a predefined expression vocabulary $\mathcal{E} = \{e_t\}_{t=1}^T$. All blendshapes share topology and vertex correspondence, enabling standard linear blendshape animation:

$$\mathbf{v}(\mathbf{w}) = \mathbf{v}^0 + \sum_{t=1}^{T} w_t(\mathbf{v}^t - \mathbf{v}^0), \tag{1}$$

where $\mathbf{v}^t$ denotes the vertex positions of $\mathcal{B}^t$ and $\mathbf{w}$ are blend weights.

RegHead addresses this problem in three stages. First, we construct semantically labeled expression observations for a predefined expression vocabulary and build a large-scale non-humanoid character expression dataset (Sec. 3.1). Second, we define a stochastic anchor-based motion representation on the neutral shape, which provides an efficient and locally expressive parameterization for non-humanoid facial deformation (Sec. 3.2). Third, given expression-specific but unregistered meshes $\{\tilde{\mathcal{M}}^t\}_{t=0}^T$, we propose a feed-forward registration network to predict anchor transformations and convert them into a corresponded blendshape basis $\{\mathcal{B}^t\}_{t=0}^T$ (Sec. 3.3). We further show our training objective and retargeting application in Sec. 3.4 and Sec. 3.5.

3.1 Expression vocabulary and dataset

A key challenge is obtaining expression observations with *consistent semantics* across identities. This challenge is particularly acute for non-humanoid heads: to our knowledge, there is no publicly available large-scale non-humanoid head blendshape dataset, and existing public 3D/4D assets [6, 16, 40] provide limited coverage in both identities and expressions. We therefore start by defining a fixed semantic expression vocabulary $\mathcal{E}$ using a small collection of high-quality, artist-rigged non-humanoid head assets.

Instead of relying on unconstrained video generation, we fine-tune an image editing model [39] using paired renderings from the artist library, enabling controlled edits from a neutral reference into each target expression in $\mathcal{E}$. Applied to text-to-image generations, this model produces semantically labeled expression image sets at scale, effectively expanding the artist-defined vocabulary from a small curated library to thousands of synthetic identities.

Given the neutral and edited expression images, we reconstruct expression-specific 3D surfaces using image-conditioned 3D/4D generation pipelines, with additional consistency heuristics to reduce inter-expression drift [36, 41]. This yields a set of raw expression meshes $\{\tilde{\mathcal{M}}^t\}_{t=0}^T$ that preserve the intended semantics but are not yet in vertex correspondence. These meshes serve as input to our feed-forward registration model (Sec. 3.3), which converts them into the final corresponded blendshape basis $\{\mathcal{B}^t\}_{t=0}^T$.

We construct a dataset of approximately 20k non-humanoid identities, each with a labeled expression set in the same semantic vocabulary. To maintain quality at scale, we additionally employ human annotators to filter low-quality

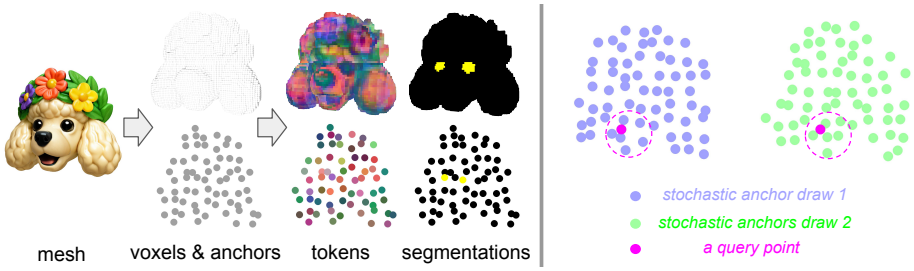

Fig. 2: (Left) We voxelize the mesh and generate stochastic anchors, then create voxel tokens including segmentation cues unprojected from multi-view renderings of the mesh. We can also obtain per-anchor tokens or features by trilinear interpolation from voxels. **(Right)** Two draws of stochastic anchors (light purple and green) and a query point (magenta) driven by neighboring anchors. Although the identity of the neighboring anchors changes when generating stochastic anchors, the blended support remains localized.

generations and reconstruction failures. Please find more dataset details in the project website.

3.2 Stochastic Anchor Motion Representation

Our goal is to convert raw expression meshes $\{\tilde{\mathcal{M}}^t\}_{t=0}^T$ into a corresponded semantic blendshape basis $\{\mathcal{B}^t\}_{t=0}^T$ by predicting, for each expression t, an anchor-based deformation from the neutral mesh to the target. Because non-humanoid facial motion is highly localized (e.g., eyelids) yet dense deformation on high-resolution surfaces is expensive, we represent motion as per-anchor transformations $\mathbf{T}^t$ defined on neutral anchors and propagate them to dense surface points via sparse-to-dense blending to obtain $\mathcal{B}^t$. This subsection defines the motion representation (anchors, voxel tokens, and propagation), while Sec. 3.3 describes how $\mathbf{T}^t$ is predicted via correspondence-aware matching.

Let $\tilde{\mathcal{M}}^0$ denote the neutral mesh for an identity. We sample a set of deformation anchors $\mathbf{A} = \{\mathbf{a}_k\}_{k=1}^K$ on $\tilde{\mathcal{M}}^0$, where each anchor acts as a local control node. A key design choice is that we *resample anchors at every training iteration* and do not optimize anchor coordinates. This stochastic anchor strategy improves local coverage and encourages the model to learn anchor-layout-invariant deformation prediction. At test time we sample anchors from the same distribution. Empirically, predictions are stable across different anchor draws. We find that using substantially denser anchor sets than seen during training does not improve fidelity and can introduce local artifacts due to a distribution shift in anchor spacing and blending weights.

We also sample a denser set of neutral query points $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^N$ from the neutral shape. The model predicts transformations only at anchors, and propagates them to queries by blending nearby anchor motions, enabling efficient yet expressive dense deformation.

For each expression mesh $\tilde{\mathcal{M}}^t$, we voxelize it and build multi-view unprojection-based voxel tokens $\mathbf{V}^t$ inspired by prior voxel-latent pipelines [42]. In our implementation, $\mathbf{V}^t$ concatenates an 8-channel voxel latent code with an unprojected semantic segmentation mask for the eye region, yielding 9-channel voxel tokens. This formulation is modular: additional 2D cues can be unprojected into the same voxel grid and appended as extra channels when available. Given any position, we obtain its feature under $\mathbf{V}^t$ by trilinear interpolation from neighboring voxels. We demonstrate this process in Fig. 2 (left).

We realize the deformation function f_θ by predicting per-anchor *pivoted similarity transforms* $\mathbf{T}^t = \{\mathcal{T}_k^t\}_{k=1}^K$ for each target expression, which is obtained by:

$$\mathbf{T}^t = f_\theta(\mathbf{A}, \mathbf{V}^t). \tag{2}$$

Then we propagate the transforms to queries via Linear Blend Skinning. For each query point $\mathbf{q}_j$, we precompute skinning weights w_{jk} via normalized inverse Mahalanobis distances from its K' nearest anchors, and deform it as

$$\hat{\mathbf{q}}_j^t = \sum_{k \in \mathcal{N}(j)} w_{jk}\, \phi(\mathbf{q}_j; \mathcal{T}_k^t, \mathbf{a}_k), \qquad \hat{\mathbf{Q}}^t = \{\hat{\mathbf{q}}_j^t\}_{j=1}^M, \tag{3}$$

where ϕ denotes applying the anchor transform to $\mathbf{q}_j$.

We demonstrate this process in Fig. 2 (right). This sparse-to-dense propagation enables localized deformations while keeping the network prediction cost linear in the number of anchors.

3.3 Feed-Forward Registration

Given the anchors $\mathbf{A}$, query points $\mathbf{Q}$, and voxel tokens $\mathbf{V}^t$ defined in Sec. 3.2, we predict per-expression anchor transforms $\mathbf{T}^t$ using correspondence-aware coarse-to-fine matching.

Our core strategy is *local feature matching* in voxel space: for each anchor, we compare its neutral token to voxel tokens within a nearby neighborhood of the target expression to estimate a local transformation. This approach relies on the locality assumption that the true correspondence of an anchor lies within the queried neighborhood. In practice, large-support motions (e.g., jaw articulation) can displace regions by more than the neighborhood radius, causing an anchor to retrieve neighbors from an unrelated facial region. Moreover, voxel tokens are expression-dependent and may be affected by reconstruction artifacts, increasing ambiguity even when the neighborhood is correct. To make local matching reliable, we adopt a coarse-to-fine design: a global matcher first predicts a coarse deformation to bring anchors into the correct basin, after which a structured local matcher refines fine-grained localized motion using validity-aware voxel neighborhoods. We demonstrate our feed-forward registration method in Fig. 3.

We represent anchors and voxel neighborhoods as feature tokens to enable attention-based matching. For each anchor $\mathbf{a}_k$, we form a neutral anchor token $\mathbf{h}_k^0 \in \mathbb{R}^d$ by encoding its position together with the interpolated neutral voxel tokens. For each target expression t, we additionally construct a set of *global*

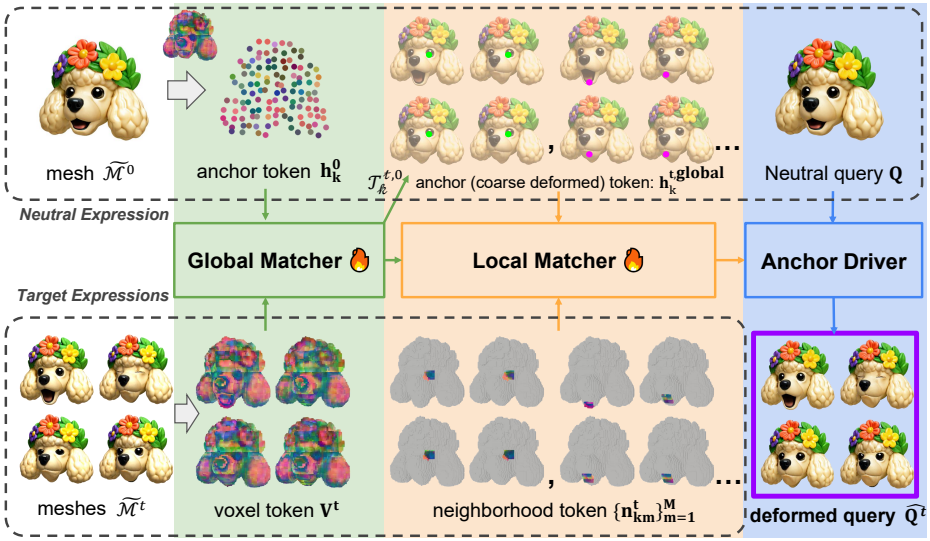

Fig. 3: Feed-Forward Registration. The goal of our feed-forward registration is to learn a deformation function f_θ that deforms the neutral expression to multiple target expressions. We split all expressions of any identity into a neutral expression and target expressions and obtain anchor and voxel tokens as described in Sec. 3.2. Then, a global matcher first predicts a coarse deformation $\mathcal{T}_k^{t,0}$ to bring anchors into a better local matching basin, and a structured local matcher then refines fine-grained localized motion using validity-aware voxel neighborhoods. As marked in orange, we show two coarsely deformed anchors $\mathbf{a}_k^t$ (green and magenta) matching to their neighborhood tokens $\{\mathbf{n}_{km}^t\}_{m=1}^M$ shown at the bottom. Finally, the local matcher predicts the per-anchor transformations that drive the neutral query.

tokens $\mathbf{v}^t$ by randomly subsampling the valid voxels from the target feature field $\mathbf{V}^t$ and encoding each voxel's position and feature. These global tokens provide long-range context for coarse alignment.

We first perform a global matching stage that aggregates long-range evidence from the global tokens into each anchor token via cross-attention:

$$\mathbf{h}_k^{t,\text{global}} = \text{XAttn}\big(\mathbf{h}_k^0, \mathbf{v}^t\big)\,. \tag{4}$$

From $\mathbf{h}_k^{t,\text{global}}$, we predict a coarse anchor transform $\mathcal{T}_k^{t,0}$ and warp the anchors to obtain coarse aligned anchor positions $\{\mathbf{a}_k^t\}$. This stage provides a robust initialization for local refinement by improving the neighborhood quality for subsequent matching.

For local refinement, we gather a structured stencil neighborhood around each coarsely warped anchor $\mathbf{a}_k^t$ in $\mathbf{V}^t$ and encode each neighbor voxel into a token $\mathbf{n}_{km}^t$ using its relative offset and voxel tokens. Instead of selecting neighbors by kNN in voxel space, we use a structured multi-radius stencil centered at each coarsely warped anchor $\mathbf{a}_k^t$ in the voxel tokens $\mathbf{V}^t$. We sample offsets on thin shell bands at multiple radii and cap the number of offsets per radius to keep

a fixed token budget. We further use the voxel validity mask to discard invalid locations, and optionally snap invalid stencil slots to nearby valid voxels to avoid empty neighborhoods. Then, we encode each neighbor voxel into a token using its relative offset and voxel tokens, yielding neighborhood tokens $\{\mathbf{n}_{km}^t\}_{m=1}^M$. Finally, we refine each anchor token by local cross-attention:

$$\mathbf{h}_k^{t,\mathrm{loc}} = \mathrm{LocalXAttn}\Big(\mathbf{h}_k^{t,\mathrm{global}}, \{\mathbf{n}_{km}^t\}_{m=1}^M\Big), \tag{5}$$

followed by lightweight anchor-graph self-attention to encourage spatially coherent motion. Finally, we predict a residual transform $\Delta\mathcal{T}_k^t$ and compose it with the coarse transform: $\mathcal{T}_k^t = \Delta\mathcal{T}_k^t \circ \mathcal{T}_k^{t,0}$. As shown in Sec. 4.3, this structured local matching stage is the primary contributor to fine expression deformation.

3.4 Training Objectives

Our training objective supervises the predicted deformation without requiring explicit point-wise correspondence between the neutral and target expressions. Given neutral query points $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^N$ sampled on the neutral mesh and their predicted deformed positions $\hat{\mathbf{Q}}^t = \{\hat{\mathbf{q}}_j^t\}_{j=1}^N$ for expression t (Sec. 3.2), we optimize a differentiable rendering loss that compares $\hat{\mathbf{Q}}^t$ with the target expression appearance, together with an auxiliary geometric loss that stabilizes the global matching stage.

For each expression mesh $\tilde{\mathcal{M}}^t$, we sample a dense set of near-surface target points $\mathbf{P}^t = \{\mathbf{p}_i^t\}_{i=1}^{N_P}$. Each point carries its RGB color and additional segmentation obtained by interpolating the corresponding voxel channels from $\mathbf{V}^t$ (Sec. 3.2). For the neutral query points $\mathbf{Q}$, we assign point attributes once from the neutral feature field $\mathbf{V}^0$ and transport them to $\hat{\mathbf{Q}}^t$.

We supervise deformation using a point-based differentiable renderer. Specifically, we render both the predicted points $\hat{\mathbf{Q}}^t$ and the target points $\mathbf{P}^t$ as isotropic Gaussian splats under randomly sampled camera views π. Compared to mesh-based rendering supervision, this point-based formulation provides stable gradients. We compute both an RGB rendering loss and a segmentation rendering loss:

$$\mathcal{L}_{\mathrm{rend}} = \mathbb{E}_{\pi\sim\mathcal{D}_{\mathrm{cam}}}\Big[\mathcal{L}_{\mathrm{rgb}}(\pi) + \lambda_{\mathrm{seg}}\,\mathcal{L}_{\mathrm{seg}}(\pi)\Big], \tag{6}$$

$$\mathcal{L}_{\mathrm{rgb}}(\pi) = \Big\|\mathcal{R}_{\mathrm{rgb}}(\hat{\mathbf{Q}}^t;\pi) - \mathcal{R}_{\mathrm{rgb}}(\mathbf{P}^t;\pi)\Big\|_2^2 \tag{7}$$

$$+ \lambda_{\mathrm{lpips}}\,\mathrm{LPIPS}\Big(\mathcal{R}_{\mathrm{rgb}}(\hat{\mathbf{Q}}^t;\pi),\, \mathcal{R}_{\mathrm{rgb}}(\mathbf{P}^t;\pi)\Big). \tag{8}$$

$$\mathcal{L}_{\mathrm{seg}}(\pi) = \Big\|\mathcal{R}_{\mathrm{seg}}(\hat{\mathbf{Q}}^t;\pi) - \mathcal{R}_{\mathrm{seg}}(\mathbf{P}^t;\pi)\Big\|_2^2, \tag{9}$$

where $\mathcal{R}_{\mathrm{rgb}}$ and $\mathcal{R}_{\mathrm{seg}}$ render the RGB and segmentation channels, respectively. We only use feature loss (LPIPS) on RGB rendering. All splats use a fixed rotation and opacity; we set the isotropic scale per point based on local point density to obtain stable supervision.

To encourage the global matcher to predict a geometrically meaningful coarse deformation, we additionally supervise the coarsely deformed anchor positions $\mathbf{A}^t = \{\mathbf{a}_k^t\}_{k=1}^K$ (Sec. 3.3). Since $\mathbf{P}^t$ is dense, we uniformly subsample a smaller set $\mathbf{P}_{\text{sub}}^t \subset \mathbf{P}^t$ with the same number as the anchors and minimize a Chamfer distance:

$$\mathcal{L}_{\text{coarse}} = \text{CD}\big(\mathbf{A}^t, \mathbf{P}_{\text{sub}}^t\big)\,. \tag{10}$$

This provides a direct geometric signal to the global stage and improves neighborhood quality for subsequent local matching. The full training loss is

$$\mathcal{L} = \mathcal{L}_{\text{rend}} + \lambda_{\text{coarse}}\,\mathcal{L}_{\text{coarse}}. \tag{11}$$

3.5 Real-Time Expression Retargeting

Given a human video, we run an off-the-shelf face-tracking system to detect dense facial landmarks and fit an internal human-head 3DMM, yielding per-frame expression coefficients $\mathbf{w}^{\text{hum}} \in \mathbb{R}^D$ and head pose. We learn a small MLP g_ψ that maps tracking coefficients to the weights of our non-humanoid blendshape vocabulary without per-frame optimization:

$$\mathbf{w} = g_\psi(\mathbf{w}^{\text{hum}}). \tag{12}$$

We optimize g_ψ once offline using a combination of supervision in the tracker coefficient space and a 3D geometric consistency loss on a human face mesh driven by the tracker coefficients. In addition to the tracked expressions, we apply the tracked pose to normalized blendshapes to obtain real-time retargeting.

4 Experiments

4.1 Implementation Details

We randomly sample K=3000 anchors on the neutral mesh surface at every training iteration. We precompute N=30000 canonical query points on the neutral shape. We shuffle the order of query points each iteration during training. Each query point is driven by its K'=10 nearest anchors with fixed skinning weights (Sec. 3.2). For each identity, we normalize and align all expression meshes $\{\tilde{\mathcal{M}}^t\}$ to a shared canonical coordinate system and voxelize them into a 64^3 grid. To construct the 8-channel voxel latent, we follow Xiang et al [42]. To obtain a segmentation mask for each voxel, we render each mesh from 15 near-frontal viewpoints and unproject the multi-view features back into the voxel grid (Sec. 3.2).

We implement our method in PyTorch and train with mixed precision on 32×A100 (80GB) GPUs. Our training split contains 10k identities and the test split contains 200 identities; each identity provides 7 expressions. For each identity, we designate a fixed neutral expression. In each training iteration, we sample one identity (local batch size 1) by randomly selecting three target expressions. The model then predicts deformations from the neutral to these three targets. We optimize the feed-forward registration model using AdamW (β_1=0.9, β_2=0.999, weight decay 10^{-4}) with a learning rate of 1×10^{-4}. We train for 2000 epochs and observe convergence after approximately 1500 epochs.

	Rendered Appearance			Visible Geometry				Runtime
Method	PSNR ↑	SSIM ↑	LPIPS ↓	CD ↓	D-Err. ↓	N-Cons. ↑	Sil. IoU ↑	Time ↓
ActionMesh	22.2731	0.8989	0.0654	0.00513	0.01032	0.9103	0.9739	**∼5 s**
V2M4	22.7849	0.9056	0.0660	0.00959	0.01827	0.9039	0.9567	∼100 s
T2Bs	23.4349	0.9064	0.0568	0.00387	0.00711	0.9160	0.9851	∼1800 s
RegHead	**23.8421**	**0.9078**	**0.0551**	**0.00358**	**0.00613**	**0.9192**	**0.9865**	**∼5 s**

Table 1: Quantitative comparison with baselines. We evaluate both rendered appearance and visible geometry. Rendered appearance is measured by PSNR, SSIM, and LPIPS under the same camera views. Visible geometry is computed on rasterized first-hit surfaces from multiple views, avoiding hidden/internal mesh surfaces. CD denotes visible Chamfer distance, D-Err. denotes mean absolute depth error, N-Cons. denotes normal consistency, and Sil. IoU denotes silhouette intersection-over-union.

4.2 Baseline Comparison

We compare RegHead with representative methods that can be adapted to the task of converting an unregistered expression mesh set into an animatable, corresponded blendshape basis. Because direct prior work on semantic blendshape construction for diverse non-humanoid heads is limited, we select baselines that operate on general 3D geometry or 4D mesh sequences and can handle diverse topology and shape variation.

For a fair comparison, all methods are provided with the same per-expression target meshes $\{\tilde{\mathcal{M}}^t\}$ as input. This isolates the registration problem from differences in upstream 3D generation. We evaluate whether the predictions match the target mesh $\tilde{\mathcal{M}}^t$ under the same rendering conditions.

ActionMesh [31] is a strong feed-forward general mesh animation baseline. Its second stage predicts deformations of a reference shape from an input mesh sequence. We use the authors' official second-stage implementation.

V2M4 [4] is an optimization-based method that registers a reference mesh to per-frame meshes from monocular video. We use the official implementation.

T2Bs [22] aligns a static 3D asset to generative video outputs to obtain registered geometry [22]. We report results produced by the authors on our test set, using multi-view renderings of our expression meshes as the video input.

Quantitative Comparison We evaluate each method from two complementary perspectives. First, since the output blendshapes are ultimately used for rendering and retargeting, we measure image-space expression fidelity by rendering each predicted expression mesh and the target mesh under the same camera views, and report PSNR/SSIM/LPIPS together with runtime per identity. Second, we evaluate visible-surface geometry. To handle coordinate differences across methods, we first estimate a global neutral-to-neutral alignment for each identity and method, and apply the same transform to all expressions. We rasterize each predicted expression and the target mesh from multiple views and compare only the first-hit visible surface. Specifically, we extract visible 3D points for Chamfer distance, compare z-buffer depth and surface normals on commonly

Fig. 4: Qualitative comparison with baselines. We show a subset of expressions from our 7-expression vocabulary. Given the same neutral input, all methods deform the neutral shape to match four representative target expressions. We render each method's predicted expression alongside the target unregistered mesh for visual reference. Our method more faithfully reproduces localized expression changes while preserving overall identity and surface quality.

visible pixels, and compute silhouette IoU from the rasterized masks. This avoids hidden or internal mesh surfaces and evaluates the geometry that contributes to the rendered avatar. Table 1 reports quantitative results on expression fidelity, visible geometry, and runtime. RegHead outperforms the feed-forward baseline ActionMesh by a clear margin. Compared to optimization-based methods, V2M4 and T2Bs, RegHead attains higher rendered fidelity and better visible-surface geometry while being orders of magnitude faster.

Qualitative Comparison Fig. 4 compares our predicted blendshapes with representative baselines on the same input neutral mesh and the same target expression meshes. The selected expressions include both large-support motion (mouth closure) and subtle localized deformation (eyes), which are particularly challenging for non-humanoid heads. Overall, our method better captures fine-grained, spatially localized expression changes while maintaining coherent global structure. ActionMesh often obviously underestimates localized facial motion,

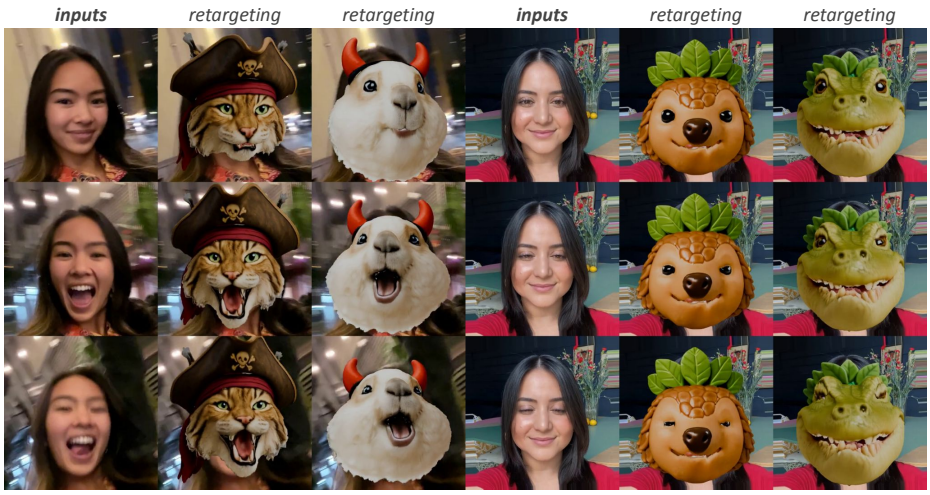

Fig. 5: Real-time retargeting from human performance to virtual characters. We show two human identities and four virtual characters driven by the same tracked performance. We show pose and expression changes across each frame. The retargeted characters reproduce both head pose and localized facial motions.

which is consistent with the absence of an explicit local matching mechanism in its deformation stage and with the domain gap between its training data and diverse non-humanoid head geometries. Optimization-based methods, V2M4 and T2Bs, produce plausible global trends but frequently exhibit artifacts or underfitting on subtle regions such as partial eyelid closure, leading to visibly mismatched expressions under the same rendering conditions.

We show additional qualitative results that demonstrate real-time retargeting from human performance as described in Sec. 3.5, and animate non-humanoid characters via our blendshapes. Fig. 5 shows retargeting results on two different human identities and four non-humanoid characters (two per identity). We include sequences dominated by pose changes and localized expression changes. Across identities and characters, the retargeted virtual characters follow both the tracked head pose and the fine-grained facial expressions, demonstrating that our blendshape vocabulary provides a stable control interface for real-time animation.

4.3 Ablation Study

We ablate key design choices in our motion representation (Sec. 3.2) and feed-forward registration model (Sec. 3.3). See the supplement for more analysis.

Ablations on stochastic anchor motion representation. Table 2a evaluates components of our motion representation. Our full model uses K=3000 stochastic shell anchors and additionally unprojects semantic segmentation cues into

Method	PSNR↑	SSIM↑	LPIPS↓
Ours	**23.842**	**0.9078**	**0.0551**
Fixed anchors	23.323	0.9044	0.0610
300 anchors	22.226	0.8987	0.0629
w/o seg.	23.391	0.9056	0.0575

(a) Motion representation (Sec. 3.2).

Method	PSNR↑	SSIM↑	LPIPS↓
Ours	**23.842**	**0.9078**	**0.0551**
w/o local	22.138	0.8963	0.0635
w/o global	23.660	0.9062	0.0577
kNN neighbors	23.055	0.9044	0.0574

(b) Registration design (Sec. 3.3).

Table 2: Ablations on stochastic anchor motion representation (a) and feed-forward registration design (b).

the voxel feature field. Replacing stochastic anchors with a fixed anchor set degrades performance with -0.52 PSNR and higher LPIPS, supporting the benefit of anchor-layout invariance induced by resampling. Reducing the anchor count to K=300 leads to a large drop across all metrics, indicating that dense anchors are important for capturing fine-scale facial motion. Finally, removing the unprojected segmentation channel (*w/o seg.*) also consistently worsens results, demonstrating that fusing lightweight 2D semantic cues into the voxel field provides useful region-aware guidance.

Ablations on feed-forward registration. Table 2b evaluates the two-stage coarse-to-fine registration design. Removing the local matching stage (*w/o local*) causes a substantial degradation, confirming that local feature matching is the primary driver of fine expression deformation. Removing the global alignment stage (*w/o global*) yields a smaller but consistent drop, suggesting that the Global matcher improves robustness by placing anchors into a better basin for subsequent local matching. Finally, replacing our validity-aware stencil neighborhood with a kNN neighborhood of the same token budget (the same number of neighbors) reduces performance, validating the benefit of structured multi-radius stencil neighborhoods for stable and informative local matching.

Stage-wise Diagnosis of Expression Generation. We provide a stage-wise visualization in Fig. 6. For each example, we show the neutral image, edited target image, raw neutral mesh, raw target expression mesh, the deformation predicted using only the global matcher, and the final RegHead result.

This analysis separates two sources of expression variation. First, the edited images and raw target meshes reveal the amplitude and locality of the upstream expression targets. Some expressions are naturally concentrated in small semantic regions, such as eyelids or mouth corners, rather than producing large global deformation. Second, comparing the global-matcher-only output with the final RegHead result shows the effect of our coarse-to-fine registration design. The global matcher captures the coarse direction of motion, but often underfits localized details. The full model, with structured local matching, recovers stronger local deformation and better matches the raw target mesh. This suggests that

Fig. 6: Stage-wise diagnosis of expression generation. Columns show the neutral image, edited target image, raw neutral mesh, raw target mesh, deformation using only the global matcher, and final RegHead result. The global matcher captures coarse expression motion, while the full model recovers stronger localized deformation.

the remaining subtle cases are largely related to localized or moderate upstream targets and the difficulty of matching fine non-humanoid facial motion.

5 Conclusion

We presented RegHead, a novel framework for constructing semantic blendshape sets for non-humanoid head avatars under a fixed expression vocabulary. We build a large-scale non-humanoid head dataset, introduce a dense stochastic anchor motion representation, and propose a fast feed-forward registration model. Experiments show that our method improves expression fidelity over baselines while running orders of magnitude faster than optimization methods, and enables real-time retargeting from human face tracking to non-humanoid characters.

Limitations. RegHead can underfit identities far from common head anatomy, such as insect-like or long-tail species with ambiguous eyes, mouths, or facial regions. These cases have weaker semantic correspondence and may require different local motion patterns for the same expression label. The final blendshapes are also affected by artifacts or identity drift in the upstream unregistered meshes.

References

1. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
2. Chen, E.M., Liu, D., Ma, S., Vasilkovsky, M., Zhou, B., Gao, Q., Wang, W., Luo, J., Metaxas, D.N., Sitzmann, V., et al.: Snapmoji: Instant generation of animatable dual-stylized avatars. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1948–1958 (2026)
3. Chen, H., Chen, X., Zhang, Y., Xu, Z., Chen, A.: Motion 3-to-4: 3d motion reconstruction for 4d synthesis. arXiv preprint arXiv:2601.14253 (2026)
4. Chen, J., Zhang, B., Tang, X., Wonka, P.: V2m4: 4d mesh animation reconstruction from a single monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11643–11653 (October 2025)
5. Chu, X., Harada, T.: Generalizable and animatable gaussian head avatar. Advances in Neural Information Processing Systems **37**, 57642–57670 (2024)
6. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., VanderBilt, E., Kembhavi, A., Vondrick, C., Gkioxari, G., Ehsani, K., Schmidt, L., Farhadi, A.: Objaverse-xl: A universe of 10m+ 3d objects. arXiv preprint arXiv:2307.05663 (2023)
7. He, Y., Gu, X., Ye, X., Xu, C., Zhao, Z., Dong, Y., Yuan, W., Dong, Z., Bo, L.: Lam: large avatar model for one-shot animatable gaussian head. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–13 (2025)
8. Jiang, Y., Zhang, L., Gao, J., Hu, W., Yao, Y.: Consistent4d: Consistent 360° dynamic object generation from monocular video. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=sPUrdFGepF
9. Jiang, Z., Zheng, C., Laina, I., Larlus, D., Vedaldi, A.: Mesh4d: 4d mesh reconstruction and tracking from monocular video. arXiv preprint arXiv:2601.05251 (2026)
10. Kaiser, J.N., Kimmel, S., Licht, E., Landwehr, E., Hemmert, F., Heuten, W.: Get real with me: Effects of avatar realism on social presence and comfort in augmented reality remote collaboration and self-disclosure. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. pp. 1–18 (2025)
11. Kang, J., Lee, S.: User experience in mobile metaverses: Nonverbal communication by avatars in zepeto. PRESENCE: Virtual and Augmented Reality **34**, 189–217 (2025)
12. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 1–14 (2023)
13. Kiuchi, A., Wieland, J., Igarashi, T., Lindlbauer, D.: Minimates: Miniature avatars for ar remote meetings within limited physical spaces. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. pp. 1–20 (2025)
14. Lewis, J.P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F.H., Deng, Z.: Practice and theory of blendshape facial models. Eurographics (State of the Art Reports) **1**(8), 2 (2014)
15. Li, H., Weise, T., Pauly, M.: Example-based facial rigging. Acm transactions on graphics (tog) **29**(4), 1–6 (2010)
16. Li, Y., Takehara, H., Taketomi, T., Zheng, B., Nießner, M.: 4dcomplete: Non-rigid motion estimation beyond the observable surface. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12706–12716 (2021)

17. Li, Z., Chen, Y., Liu, P.: Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. Advances in Neural Information Processing Systems **37**, 21377–21400 (2024)
18. LIANG, H., Yin, Y., Xu, D., hanxue liang, Wang, Z., Plataniotis, K.N., Zhao, Y., Wei, Y.: Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=grrefkWEES
19. Liu, I., Xu, Z., Yifan, W., Tan, H., Xu, Z., Wang, X., Su, H., Shi, Z.: Riganything: Template-free autoregressive rigging for diverse 3d assets. ACM Transactions on Graphics (TOG) **44**(4), 1–12 (2025)
20. Liu, S., Zhang, Y., Li, W., Lin, Z., Jia, J.: Video-p2p: Video editing with cross-attention control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8599–8608 (2024)
21. Luo, J., Liu, J., Davis, J.: Splatface: Gaussian splat face reconstruction leveraging an optimizable surface. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 774–783. IEEE (2025)
22. Luo, J., Wang, C., Vasilkovsky, M., Shakhrai, V., Liu, D., Zhuang, P., Tulyakov, S., Wonka, P., Lee, H.Y., Davis, J., et al.: T2bs: Text-to-character blendshapes via video generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13625–13637 (2025)
23. Ma, W.C., Lamarre, M., Danvoye, E., Ma, C., Ko, M., von der Pahlen, J., Wilson, C.A.: Semantically-aware blendshape rigs from facial performance measurements. In: SIGGRAPH ASIA 2016 technical briefs, pp. 1–4 (2016)
24. 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: European Conference on Computer Vision. pp. 405–421. Springer (2020)
25. Nguyen, T.H., Luo, J., Nie, Y., Li, H., Qian, G.G., Wang, J.: Ffavatar: Few-shot, feed-forward, and generalizable avatar reconstruction. arXiv preprint arXiv:2605.15320 (2026)
26. Pan, Y., Tan, S., Cheng, S., Lin, Q., Zeng, Z., Mitchell, K.: Expressive talking avatars. IEEE Transactions on Visualization and Computer Graphics **30**(5), 2538–2548 (2024)
27. Pérez, J.C., Nguyen-Phuoc, T., Cao, C., Sanakoyeu, A., Simon, T., Arbeláez, P., Ghanem, B., Thabet, A., Pumarola, A.: Styleavatar: Stylizing animatable head avatars. In: Proceedings of the ieee/cvf winter conference on applications of computer vision. pp. 8678–8687 (2024)
28. Qian, S., Kirschstein, T., Schoneveld, L., Davoli, D., Giebenhain, S., Nießner, M.: Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20299–20309 (2024)
29. Ren, J., Xie, C., Mirzaei, A., Kreis, K., Liu, Z., Torralba, A., Fidler, S., Kim, S.W., Ling, H., et al.: L4gm: Large 4d gaussian reconstruction model. Advances in Neural Information Processing Systems **37**, 56828–56858 (2024)
30. Sabathier, R., Mitra, N.J., Novotny, D.: Lim: Large interpolator model for dynamic reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6154–6164 (2025)
31. Sabathier, R., Novotny, D., Mitra, N.J., Monnier, T.: Actionmesh: Animated 3d mesh generation with temporal 3d diffusion. arXiv preprint arXiv:2601.16148 (2026)

32. Sang, S., Zhi, T., Song, G., Liu, M., Lai, C., Liu, J., Wen, X., Davis, J., Luo, L.: Agileavatar: Stylized 3d avatar creation via cascaded domain bridging. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–8 (2022)
33. Shi, Y., Liu, Y., Wu, Y., Liu, X., Zhao, C., Luo, J., Zhou, B.: Drive any mesh: 4d latent diffusion for mesh deformation from video. arXiv preprint arXiv:2506.07489 (2025)
34. Song, C., Zhang, J., Li, X., Yang, F., Chen, Y., Xu, Z., Liew, J.H., Guo, X., Liu, F., Feng, J., et al.: Magicarticulate: Make your 3d models articulation-ready. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15998–16007 (2025)
35. Song, L., Chen, L., Liu, C., Liu, P., Xu, C.: Texttoon: Real-time text toonify head avatar from single video. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
36. Team, T.H.: Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material (2025)
37. Wang, D., Meng, H., Cai, Z., Shao, Z., Liu, Q., Wang, L., Fan, M., Zhan, X., Wang, Z.: Headevolver: Text to head avatars via expressive and attribute-preserving mesh deformation. In: 2025 International Conference on 3D Vision (3DV). pp. 211–221. IEEE (2025)
38. Wang, X., Zhao, S., Wang, Y., Han, H.Z., Liu, X., Yi, X., Tong, X., Li, H.: Raise your eyebrows higher: Facilitating emotional communication in social virtual reality through region-specific facial expression exaggeration. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. pp. 1–22 (2025)
39. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324
40. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., et al.: Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 803–814 (2023)
41. Xiang, J., Chen, X., Xu, S., Wang, R., Lv, Z., Deng, Y., Zhu, H., Dong, Y., Zhao, H., Yuan, N.J., Yang, J.: Native and compact structured latents for 3d generation. Tech report (2025)
42. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. arXiv preprint arXiv:2412.01506 (2024)
43. Xiang, J., Gao, X., Guo, Y., Zhang, J.: Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1802–1812 (2024)
44. Xie, Y., Yao, C.H., Voleti, V., Jiang, H., Jampani, V.: Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. arXiv preprint arXiv:2407.17470 (2024)
45. Xu, Z., Li, Z., Dong, Z., Zhou, X., Newcombe, R., Lv, Z.: 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. arXiv preprint arXiv:2506.08015 (2025)
46. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)

47. Yao, C.H., Xie, Y., Voleti, V., Jiang, H., Jampani, V.: Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13248–13258 (2025)
48. Yoon, S., Yun, K., Seo, K., Cha, S., Yoo, J.E., Noh, J.: Lego: Leveraging a surface deformation network for animatable stylized face generation with one example. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4505–4514 (2024)
49. Zeng, Y., Jiang, Y., Zhu, S., Lu, Y., Lin, Y., Zhu, H., Hu, W., Cao, X., Yao, Y.: Stag4d: Spatial-temporal anchored generative 4d gaussians. In: European Conference on Computer Vision. pp. 163–179. Springer (2024)
50. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. ACM Transactions On Graphics (TOG) **42**(4), 1–16 (2023)
51. Zhang, H., Luo, J., Wan, B., Zhao, Y., Li, Z., Vasilkovsky, M., Wang, C., Wang, J., Ahuja, N., Zhou, B.: Rigmo: Unifying rig and motion learning for generative animation. arXiv preprint arXiv:2601.06378 (2026)
52. Zhang, J.P., Pu, C.F., Guo, M.H., Cao, Y.P., Hu, S.M.: One model to rig them all: Diverse skeleton rigging with unirig. ACM Transactions on Graphics (TOG) **44**(4), 1–18 (2025)