RIG, RENDER, REFINE: AGENTIC 3D RIGGING THROUGH ITERATIVE FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

We present an agent-based approach to 3D mesh rigging and animation that frames rigging as an iterative cycle of perception, construction, and verification. Existing methods typically predict skeletons and skinning weights directly from 3D geometry, but geometry alone may not determine how an object should articulate or whether a proposed rig will produce plausible deformations under motion. Our key idea is to inspect a rig's behavior under articulation and use these observations to guide its refinement. Starting from a static mesh, we construct and iteratively refine a 3D skeleton by integrating multiview renderings, geometric structure, mesh-part information, and 2D skeleton proposals. We then generate skinning weights and evaluate the resulting rig under diagnostic poses, using observed deformation errors to refine both the weights and the skeleton. Experiments on a challenging dataset of structurally diverse 3D assets demonstrate improved skeleton and deformation accuracy over state-of-the-art rigging methods.

1 INTRODUCTION

Rigging is a key step in transforming 3D assets into animation-ready content for applications including film and games, virtual and augmented reality, robotics, simulation, and digital content creation. In practice, rigging requires substantial manual effort: artists must identify articulated parts, place and connect joints, define skeletal hierarchies, and assign skinning weights before an asset can be animated. While established tools and category-specific templates can accelerate this process, creating rigs for diverse 3D assets remains labor intensive, particularly when objects have unusual structures or fall outside of common categories.

Recent work has addressed automated rigging as a learned prediction problem, moving from category-specific or template-based methods toward models that directly infer skeletons and skinning weights from 3D assets like RigNet Xu et al. (2020). More recent transformer- and diffusion-based approaches extend this paradigm to more diverse object geometries and reduce reliance on predefined templates such as RigAnything Liu et al. (2025) and Puppeteer Song et al. (2025a). However, these approaches primarily operate as feed-forward predictors: they infer a rig from the available input, but do not explicitly evaluate how the predicted rig behaves under motion and use that behavior to revise the construction. This is a key limitation for rigging, where errors in the resulting rig may only become apparent after inspecting the constructed skeleton and observing the mesh under articulation.

Our key insight is that rigging should instead be treated as an iterative process of perception, construction, and verification. We build an agentic system that reasons about how an object should articulate, constructs a rig, inspects its behavior under articulation, and uses these observations to determine the necessary modifications. This process integrates complementary evidence: (i) 2D visual priors provide semantic cues about parts and their intended articulation, (ii) 3D geometry constrains the rig spatially, and (iii) animation provides evidence about how the proposed rig behaves under different articulations.

Our approach consists of two stages connected by agent-based refinement. First, we design a *mesh-to-skeleton* stage, which combines multiview renderings, medial geometry, mesh-part decompositions when available, and skeleton proposals from 2D generative models to construct a 3D skeleton. The resulting skeleton is rendered and reviewed to iteratively refine joint placement and connectivity. Second, we design a *skeleton-to-skinning* stage generates skinning weights and evaluates the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

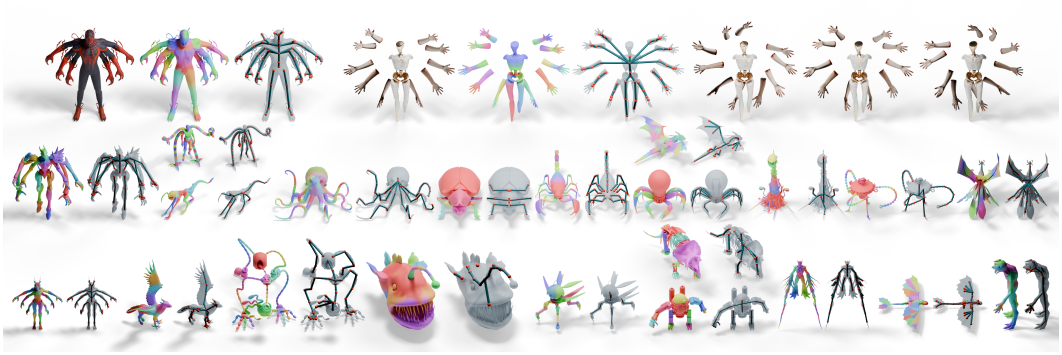


Figure 1: **Results of our approach, ReRig-Agent.** We introduce an agentic approach that transforms static meshes into animation-ready assets. Examples show predicted skeletons, color-coded skinning weights, and posed meshes across diverse objects with uncommon articulation structures.

resulting rig under articulation. Diagnostic poses expose local deformation problems which provide evidence for evaluating the skinning weights and motion. By inspecting the animated mesh, the system can refine the skinning weights and motion, while structural errors can trigger further skeleton refinement. The final output is an animation-ready 3D asset with a refined skeleton, skinning weights, and motion sequences.

Our contributions can be summarized as follows:

- We propose an agentic approach that treats 3D rigging as an iterative process of perception, construction, and verification, using the behavior of the resulting rig to guide subsequent refinement.
- We integrate complementary semantic, geometric, and animation evidence for rigging, leveraging multiview renderings, medial geometry, mesh-part information and diagnostic deformations.
- We evaluate our approach on a dataset of structurally diverse 3D assets and demonstrate improved skeleton and deformation accuracy compared with state-of-the-art rigging methods.

2 Related Work

2.1 Rigging of Static Shapes

An important line of papers studies how to rig one static shape. The pioneering work Pinocchio Baran & Popović (2007) automatically rigs a static 3D character mesh by fitting a predefined skeleton to its geometry and computing skinning weights. Several other papers studied how to compute curve skeletons of a 3D shape as meshes Au et al. (2008); Tagliasacchi et al. (2012) and point clouds Huang et al. (2013), which support rigging applications.

Recent papers explore learning-based rigging techniques that output a skeleton and associate it with rigging weights through a neural network. RigNet Xu et al. (2020) is a foundational work that offers category-general prediction of joint placement, skeleton topology, and skinning weights from a mesh. UniRig Zhang et al. (2025a), RigAnything Liu et al. (2025), MagicArticulate Song et al. (2025b), and Puppeteer Song et al. (2025a) adopt autoregressive skeleton generation, coupled respectively with attention-based skinning, diffusion-based joint prediction and connectivity/skinning heads, and functional-diffusion skinning. Auto-Connect Guo et al. (2025) strengthens autoregressive prediction through connectivity-preserving representations and topology-aware preference optimization, while TokenRig peng Zhang et al. (2026) unifies skeleton and skinning prediction in a single autoregressive token sequence. In contrast, Anymate Deng et al. (2025) explores regression-, diffusion-, and volume-based predictors within a staged rigging pipeline, and HumanRig Chu et al. (2025) uses prior-guided, coarse-to-fine regression with mesh–skeleton attention for fixed-topology humanoids. These methods primarily learn from large collections of rig-annotated assets. They do not leverage motion data, that rigged models should fit and therefore should be coupled with computing the skeleton and associated rigging weights.

2.2 Joint Rigging and Motion Prediction

The idea of extracting skeletons and rigging weights from different poses of a shape dates back to early work of skinning mesh animations James & Twigg (2005), which introduced mean-shift clustering of local rigid transformations associated with each vertex to obtain bones. This approach is later improved in example-based skeleton extraction Schaefer & Yuksel (2007). Smooth skinning decomposition with rigid bones Le & Deng (2012) and follow-by work Le & Deng (2014) introduced an alternating optimization approach to refine bone transformations and rigging weights.

Regarding learning-based approaches, MoRig Xu et al. (2022) uses motion cues from point-cloud sequences to predict skeletons and skinning weights, followed by motion transfer, while RigMo Zhang et al. (2026a) jointly learns Gaussian-bone rigs and temporal transformations from mesh sequences without rig annotations. ACT Long et al. (2026) more closely integrates rigging and animation through a diffusion transformer that models skeletal structure and motion in a shared latent space, conditioned on VLM-derived topology priors. Its geometry-guided decoder couples the generated kinematics with surface deformation, unifying rigging, skinning, and motion synthesis. In contrast, our approach emphasizes an agent-driven construction-and-refinement process: agents coordinate geometric tools and semantic cues, inspect diagnostic poses and rendered animations, and iteratively refine skeletons, skinning weights, and motion through visual feedback.

The second of our pipeline is related to R-DMesh Wu et al. (2026), which predicts video-guided vertex trajectories, and TopoCap Pu et al. (2026), which generates skeletal motion conditioned on a supplied rig. In contrast, we adopt an agent-driven refinement loop: agents coordinate motion fitting, inspect diagnostic poses and rendered animations, and iteratively revise skinning weights, motion, and, when necessary, the skeleton—using animation as visual feedback for rig refinement.

2.3 Agentic Systems for 3D and 4D Content Creation

Recent work explores LLM- and VLM-based agents for diverse 3D and 4D tasks. LL3M Lu et al. (2025) and ShapeCraft Zhang et al. (2025b) construct structured 3D objects through procedural modeling, while SceneCraft Hu et al. (2024) and SceneWeaver Yang et al. (2025) combine spatial planning, graphics tools, and iterative feedback for scene generation. Beyond static content, Articulate-Anything Le et al. (2025) constructs articulated objects through program generation and visual critique. Articraft Zhou et al. (2026) scales articulated asset generation through a specialized programming interface for defining parts, geometry, and joints, with structured validation feedback. GS-Agent Zhang et al. (2026b) further integrates physics simulation to generate dynamic 4D worlds. Complementing these systems, Instruct-Particulate Li et al. (2026) predicts part segmentation and joint parameters from meshes conditioned on kinematic specifications, which can be supplied by VLMs. Together, these works demonstrate how semantic reasoning can guide specialized geometric tools and use feedback to refine generated assets. Our approach builds on this paradigm for rigging, using agents to integrate 2D semantic cues with 3D geometry and, importantly, to inspect the resulting rig under articulation and use its behavior to refine the skeleton and skinning weights.

3 Technical Approach

We introduce an agent-based approach that transforms a static mesh into an animation-ready asset through two stages: (1) mesh-to-skeleton and (2) skeleton-to-skinning (Fig. 2). In both stages, agents iteratively propose, evaluate, and refine their outputs using geometric and visual feedback. The central idea is to use agents to integrate complementary geometric, semantic, and visual evidence, and moreover, use the resulting rig behavior as feedback for subsequent refinement.

3.1 Mesh-to-Skeleton

Given a static mesh $\mathcal{M}$, the goal of this stage is to construct a skeleton $\mathcal{S} = (\mathbf{J}, \mathbf{p})$. Here, $\mathbf{J} \in \mathbb{R}^{K\times 3}$ are the joint positions, and $\mathbf{p} = (p_1, \ldots, p_K)$ encodes the joint hierarchy, where p_i denotes the index of the parent of joint i. The number of joints K is determined during construction. Constructing the skeleton requires deciding which parts should move together and where relative motion should occur. Geometric shape alone does not resolve these decisions. A curved horn and a flexible tail may both form elongated branches, but they require different control structures. We therefore provide our

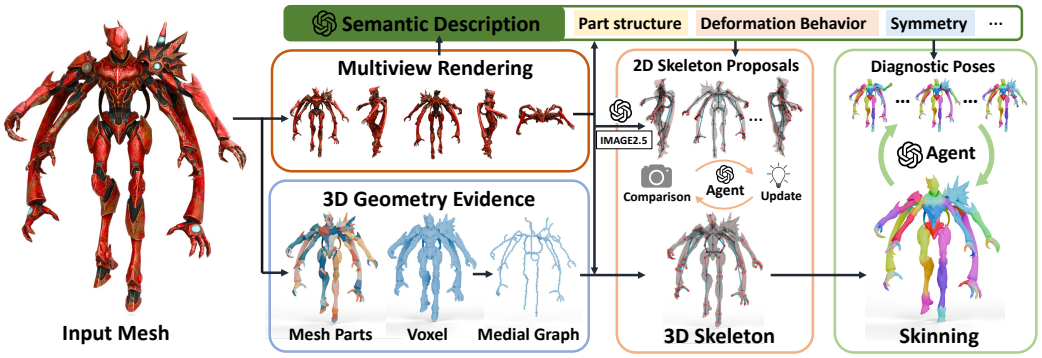


Figure 2: **Overview of ReRig-Agent.** Our agentic system first interprets multiview renderings and 3D geometry to form a semantic description of the object. This understanding guides iterative refinement in two stages: skeleton construction using 2D skeleton proposals and 3D verification, followed by skinning and articulation updates using diagnostic pose feedback.

agentic system with multiple sources of visual and geometric evidence for skeleton construction. After the initial construction, the system can revise the skeleton through multiview verification.

3.1.1 PERCEPTION

Multiview rendering. First, we provide the agentic system to multiview renderings $\mathcal{I} = \{I_v\}_{v=1}^{N}$ of the original mesh. These images reveal visible parts, their spatial relationships, and possible articulation regions. An attachment identifies where one part connects to another. Potential articulation points indicate where connected parts may rotate or bend relative to each other. The agent also identifies regions that should move as rigid units and regions that may deform.

Medial graph extraction. Starting from the original mesh, we perform voxelization, fill its interior, and apply the 3D thinning algorithm of Lee et al. (1994). The retained voxel centers form a medial graph $\mathcal{G} = (\mathbf{X}, \mathcal{E}_g)$, with edges connecting each voxel to its neighbors in the $3 \times 3 \times 3$ surrounding region. This graph acts as a geometric scaffold for potential skeletal chains. We provide implementation details in Appendix A.1.

Mesh-part decomposition. When available, mesh connectivity and existing scene or primitive boundaries provide candidate part groups $\mathcal{P} = \{P_k\}$. The agent reviews each group separately and within the assembled object. The agentic system can use meaningful groups as additional evidence and does not force every geometric group to become a separate skeletal part.

Semantic description. Given the previous information, we ask the agentic system to summarize it into a semantic description $\mathcal{H}$. For each functional part, $\mathcal{H}$ records its role, corresponding mesh regions and branches, attachments, potential articulation points, and expected rigid or flexible behavior. It also records supported symmetry and unresolved ambiguities. These descriptions guide joint placement, while part attachments inform the skeleton's parent relationships.

2D skeleton proposals. Next, we condition an image generative model with each of the original mesh renderings I_v and a design prompt $q(\mathcal{H})$ derived from the semantic description. The prompt describes the identified functional parts, their attachments, and the intended articulation. It also identifies rigid regions that should not receive unnecessary bending controls. For each rendering I_v, the output is an image D_v of the proposed skeleton design over the rendered object:

$$D_v = \mathrm{Gen}\big(I_v, q(\mathcal{H})\big), \qquad \mathcal{D} = \{D_v\}_{v=1}^{N}. \tag{1}$$

The collection $\mathcal{D}$ provides visual articulation proposals from multiple views. The agentic system reviews these proposals and may reject proposed joints or connections that are inconsistent with its semantic interpretation, e.g., additional joints on a part that has been identified as rigid. Here, for the image generative model Gen, we use GPT Image 2.5 OpenAI.

3.1.2 CONSTRUCTION

Using $\mathcal{H}$, $\mathcal{G}$, and $\mathcal{D}$, the agentic system decides which joints are needed, which parts they control, and how they are connected. These semantic decisions define the intended articulation structure before determining the joint coordinates.

For each proposed joint, the agent identifies corresponding landmarks in the original calibrated views and an appropriate location on the medial graph. We introduce a bounded fitting procedure which determines the 3D joint location by balancing reprojection agreement, proximity to the medial graph, and soft interior constraints. The resulting skeleton $\mathcal{S}_0$ is validated for joint placement, parent assignments, and overall hierarchy. The system additionally verifies that each skeletal chain corresponds to the intended part and attaches to the correct region.

3.1.3 VERIFICATION

We overlay the current skeleton $\mathcal{S}_t$ on multiple mesh views to obtain $\mathcal{I}_t$. The VLM inspects these renderings together with the semantic description $\mathcal{H}$, medial graph $\mathcal{G}$, and refinement history $\mathcal{R}_t$, which records previous edits and their review outcomes. It selects a geometric editing function f_t and arguments θ_t specifying the affected joints, placement landmarks, and parent assignments. The function converts landmarks into 3D positions when needed and updates the skeleton while preserving unaffected joints:

$$(f_t, \theta_t) = \mathrm{VLM}(\mathcal{S}_t, \mathcal{I}_t, \mathcal{H}, \mathcal{G}, \mathcal{R}_t), \qquad \widetilde{\mathcal{S}}_{t+1} = f_t(\mathcal{S}_t; \theta_t). \tag{2}$$

The edit and its review outcome are added to the history for the next iteration. Details and pseudocode are provided in Appendix A.3 and Algorithm 1.

After the edit, the new candidate skeleton is rendered from the same views and compared with the previous skeleton. The agent evaluates whether the proposed edit resolves the identified problem without introducing new errors. Accepted edits update the skeleton, while rejected edits leave it unchanged. This process is repeated until no further improvement is identified or the iteration budget is exhausted. The resulting skeleton $\mathcal{S}^\star$, together with the semantic description $\mathcal{H}$, is then passed to the skinning and animation stage.

3.2 SKELETON-TO-SKINNING

This stage takes the mesh $\mathcal{M}$, the reviewed skeleton $\mathcal{S}^\star$, and the semantic description $\mathcal{H}$ as input, and produces skinning weights $\mathbf{W} \in \mathbb{R}^{n \times B}$, where n is the number of mesh vertices and B the number of deforming bones. The system first initializes the skinning weights using the geometric and semantic evidence, and then refines them by inspecting the resulting deformations. For these deformations, we use diagnostic poses that can isolate the effect of individual articulations. During pose fitting and weight refinement, the skeleton structure remains fixed and potential changes to joint positions or hierarchy are handled by the skeleton refinement stage.

3.2.1 SEMANTIC SKINNING INITIALIZATION

Deformation regions. The agentic system uses $\mathcal{H}$ to identify rigid parts, joint transitions, and flexible regions. For each region, it determines which bones may influence its vertices and where blending is appropriate. Mesh-part groups provide additional evidence for these decisions, but geometric separation alone does not determine motion, since disconnected components may still follow the same bone, while a single connected surface may span multiple articulations.

Geometry-supported weights. Our method samples deforming bones within the occupied mesh volume and propagates distances through its interior, following the principle of geodesic voxel binding Dionne & de Lasa (2013). These distances provide geometric support for assigning bone influences without relying only on Euclidean proximity. The agent restricts influences according to part ownership and selects blending widths based on local geometry. Rigid attachments receive fixed assignments, while joint neighborhoods blend between the relevant bones. Local interpolation and smoothing produce an initial weight matrix $\mathbf{W}_0$. Details of interior distance computation, influence assignment, and smoothing are provided in Appendix A.2.

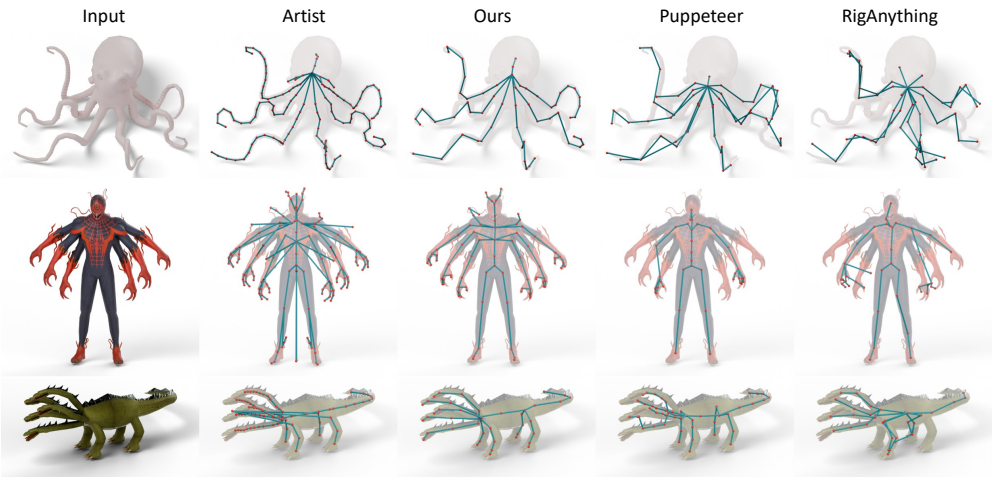


Figure 3: **Qualitative comparison of skeletons on objects with uncommon structures.** Our predictions follow the octopus tentacles and preserve separate chains for the additional arms and multiple necks. The baselines show missing branches or chains that deviate from the corresponding parts.

3.2.2 Diagnostic-Pose Refinement

The initialized weights may appear plausible in the rest pose but often produce undesirable deformations during articulation. To identify such failures, we construct a set of diagnostic poses $\mathcal{Q} = \{\boldsymbol{\theta}^d\}_{d=1}^{D}$ based on the skeleton $\mathcal{S}^\star$ and description $\mathcal{H}$. These poses exercise individual joints and articulated chains under controlled motions, making local deformation errors easier to identify.

For each diagnostic pose, the system inspects multiview renderings of the deformed mesh together with the skeleton and skinning weights. We additionally compute four geometric diagnostics: edge distortion E_{edge}, core-following error E_{core}, unrelated-region drift E_{drift}, and triangle collapse E_{collapse}. We combine them into the diagnostic score:

$$E_{\text{diag}}(\mathbf{W}; \mathcal{S}^\star, \mathcal{Q}) = w_{\text{edge}}E_{\text{edge}} + w_{\text{core}}E_{\text{core}} + w_{\text{drift}}E_{\text{drift}} + w_{\text{collapse}}E_{\text{collapse}}. \tag{3}$$

The individual terms measure surface distortion, whether rigid regions follow their assigned bones, unintended motion outside the active articulation, and severe loss of triangle area, respectively. We provide precise term definitions and weights in Appendix B

Based on the rendered observations and diagnostic scores, at refinement round r, the VLM inspects rendered observations $\mathcal{O}_r$, diagnostic scores $\mathbf{E}_r$, and the semantic description $\mathcal{H}$ together with the weight-refinement history $\mathcal{R}_r^W$. It selects a weight-editing function f_r^W and arguments θ_r^W specifying the affected vertices, relevant bones, and changes to ownership or blending. The function produces candidate weights while keeping the skeleton $\mathcal{S}^\star$ fixed:

$$(f_r^W, \theta_r^W) = \text{VLM}(\mathbf{W}_r, \mathcal{S}^\star, \mathcal{H}, \mathcal{O}_r, \mathbf{E}_r, \mathcal{R}_r^W), \qquad \widehat{\mathbf{W}}_{r+1} = f_r^W(\mathbf{W}_r; \theta_r^W). \tag{4}$$

The current weights $\mathbf{W}_r$ and candidate weights $\widehat{\mathbf{W}}_{r+1}$ are evaluated under identical poses and views. The candidate becomes $\mathbf{W}_{r+1}$ if it improves the diagnostic score and targeted deformation without introducing new failures; otherwise, $\mathbf{W}_r$ is retained. Keeping the skeleton and poses fixed isolates the effect of the weight update. The proposed edit and its review outcome are recorded in $\mathcal{R}_r^W$ for the next iteration. Refinement stops when no supported improvement remains or the iteration budget is exhausted. Errors involving pivots, missing articulations, or hierarchy are passed back to skeleton refinement. Details and pseudocode are provided in Appendix B.4 and Algorithm 2.

4 Experiments

In our experiments, we evaluate the skeleton and skinning quality of our approach against state-of-the-art rigging methods and ablate its main components. We further present a user study and qualitative results demonstrating the behavior of our approach across diverse 3D assets.

	Skeleton				Deformation			
Method	J2J ↓	J2B ↓	B2B ↓	Count ↓	ADE ↓	MDE ↓	CD ↓	ELS ↑
Puppeteer Song et al. (2025a)	.05951	.04844	.09439	111.80	.02343	.14706	.00972	.80527
RigAnything Liu et al. (2025)	.06559	.05498	.10248	115.98	.02309	.13686	.01015	-14.30589
Ours	**.04022**	**.03012**	**.08205**	**95.50**	**.01306**	**.08573**	**.00582**	**.92979**

Table 1: **Quantitative evaluation of the complete rig.** We report result on our curated dataset. Deformation is evaluated by fitting joint rotations to the reference motion while keeping the predicted skeleton and skinning weights fixed. Count denotes absolute joint-count error.

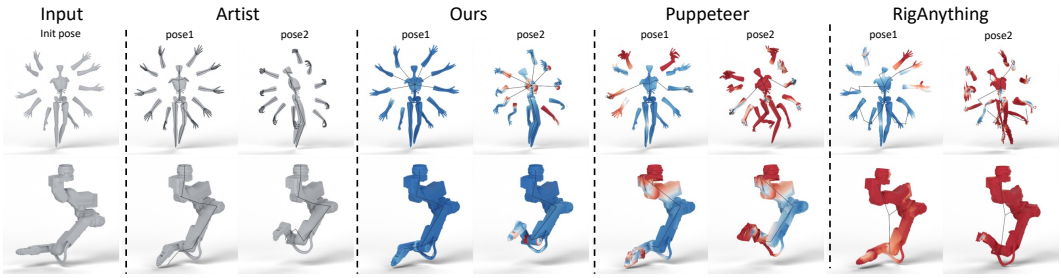


Figure 4: **Deformation comparison.** We show two meshes under two reference/artist poses each. Each method uses its predicted skeleton and skinning weights. Blue indicates lower vertex error and red indicates higher error. ReRig-Agent more closely reproduces the reference deformations.

4.1 Experimental Setup

Evaluation dataset. Existing rigging approaches are particularly challenged by uncommon morphologies. We therefore curate 50 rigged objects from Sketchfab to test generalization beyond common humanoid and quadruped structures. The collection includes multi-limbed creatures, unusual body layouts, and mechanical assemblies. Artist animations provide reference motion when available; otherwise, a reviewed synthetic sequence is constructed on the artist rig.

Evaluation protocol. For our evaluation with the state-of-the-art, we choose two of the most recent approaches for automated rigging that provide official public implementations, Puppeteer Song et al. (2025a) and RigAnything Liu et al. (2025). We compare ReRig-Agent with them under two settings. With *mesh-only input*, each method predicts a skeleton and skinning weights. To evaluate the resulting rig, a common evaluator fits joint rotations to match the reference animated surface while keeping the predicted rig fixed. This evaluates how well the rig can reproduce the reference deformation, With a *shared artist skeleton*, each method predicts only the skinning weights and is evaluated under identical joint motion without fitting.

Each object is evaluated at ten shared reference poses. RigAnything requires a supplied-skeleton adapter that supports 13 objects of our curated dataset, so comparisons involving this adapter evaluate all methods using the same *13-object subset*. Reference weights are used only for evaluation. We provide sampling procedures, binding conventions, and baseline adaptations in Appendix D.

Evaluation metrics. Following RigAnything Liu et al. (2025) and Puppeteer Song et al. (2025a), skeleton quality is measured by joint-to-joint (J2J), joint-to-bone (J2B), and bone-to-bone (B2B) distances, while skinning weights under the shared artist skeleton are compared using influence precision, recall, and vertex-wise L_1 error. Joint-count error is also reported. Deformation quality is evaluated using average vertex error (ADE), maximum vertex error (MDE), nearest-vertex distance (CD), and edge-length agreement (ELS). All distances are normalized by mesh size. Deformation scores are averaged over poses, and dataset-level results are averaged equally across objects. We provide detailed metric definitions, reference-motion fitting, and aggregation protocols in Appendix D.

4.2 Complete-Rig Evaluation

Skeleton geometry. Table 1 compares predicted skeletons with artist rigs for our dataset. ReRig-Agent achieves the lowest mean errors across all reported skeleton metrics. Its J2J error is 32.4%

Dataset	Method	Deformation				Skinning weights		
		ADE ↓	MDE ↓	CD ↓	ELS ↑	Precision ↑	Recall ↑	L_1 ↓
50-object set	Puppeteer Song et al. (2025a)	.00555	.12919	.00227	.47591	**61.40**	66.59	.81661
	Ours	**.00411**	**.11726**	**.00190**	**.91989**	59.12	**72.74**	**.72246**
13-object set	Puppeteer Song et al. (2025a)	.00322	.07580	.00171	.94646	71.13	69.66	.64134
	RigAnything Liu et al. (2025)	.00946	.12113	.00517	.92719	22.99	**92.01**	1.03390
	Ours	**.00249**	**.04153**	**.00139**	**.97065**	**71.50**	78.15	**.51596**

Table 2: **Quantitative evaluation of skinning.** We report results on our curated dataset. We use a shared artist skeleton. Precision and recall are percentages.

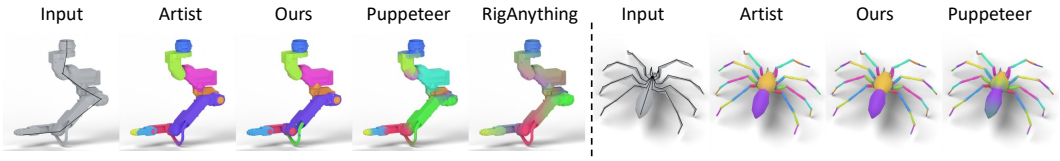


Figure 5: **Skinning-weight comparison.** All methods use the same artist skeleton. Colors indicate bone influences and blending across the mesh. Our method better matches the artist's part assignments.

lower than Puppeteer and 38.7% lower than RigAnything. These metrics measure geometric agreement with the artist rigs rather than articulation quality alone.

Animated deformation. ReRig-Agent reduces mean ADE by 44.3% relative to Puppeteer and 43.4% relative to RigAnything. It also achieves lower MDE and CD and higher ELS (Table 2). These results evaluate the complete predicted rig and therefore reflect the combined effect of skeleton and skinning quality. Figure 4 compares the animated surfaces at matched poses and viewpoints, complementing the quantitative evaluation.

4.3 Skinning with a Shared Skeleton

For this experiment, every method receives the same artist skeleton and is evaluated under identical joint motion, isolating the effect of skinning weights. Table 2 reports deformation and weight agreement. Across all 50 objects, ReRig-Agent reduces ADE by 26.0% and L_1 error on skinning weights by 11.5% relative to Puppeteer. On the 13-object subset, the reductions relative on the same metrics compared to adapted RigAnything are 73.6% and 50.1% respectively. Precision and recall show complementary tradeoffs: Puppeteer is competitive with us on precision, while RigAnything has higher recall, but much lower precision. Figure 5 compares how the predicted influences align with object parts and blend near articulations.

4.4 Ablation Studies

In this section, we present two ablations focusing on the effect of iterative refinement. In Appendices E.2 and E.4, we include further discussion and ablations of our system.

Method	J2J ↓	J2B ↓	B2B ↓	Count ↓
W/o refinement	.04223	.03180	.08349	96.00
Ours	**.04022**	**.03012**	**.08205**	**95.50**

Table 3: **Effect of skeleton refinement.** We report results on our curated dataset. We compare our method's initial skeletons with the final estimates after iterative refinement.

Effect of skeleton refinement. We perform an ablation where we evaluate our pipeline without and with the skeleton refinement. We report results in Table 3. We observe that refinement consistently improves results, reducing J2J and J2B errors by 4.74% and 5.27% respectively. We note that these metrics measure agreement with artist-designed rigs, which sometimes can be less detailed than our results (e.g., ignoring finger or toe articulation which is captured by our approach). We complement the quantitative evaluation with visual comparisons and per-object statistics in Appendix E.1.

Effect of diagnostic-pose refinement on deformation quality. We perform an ablation where we evaluate our pipeline without and with refinement based on diagnostic poses. We present results in Figure 6. We observe improvements in most metrics (e.g., 6.81% reduction in MDE) and only a

Diagnostic Pose Refinement	ADE ↓	MDE ↓	CD ↓	ELS ↑
Without	.01310	.09199	**.00579**	.91449
With	**.01306**	**.08573**	.00582	**.92979**



Figure 6: **Effect of diagnostic pose refinement.** We show results with and without refinement based on diagnostic poses. Left: Quantitative results on our curated dataset show improvement across most metrics with only a slight degradation in CD. Right: Artist deformation and our results with and without diagnostic refinement under the same pose. Colors indicate vertex position error relative to the artist reference, from blue (low) to red (high). Without refinement, arm motion incorrectly deforms mesh regions outside the arm, causing severe stretching. Diagnostic feedback corrects these unintended skinning influences and reduces the resulting artifacts.

small degradation for CD. Qualitative results also support this refinement. We provide additional diagnostic measurements, visual comparisons, and remaining failure cases in Appendix E.3.

5 Applications

Video-guided skeletal animation. Our system can also be used to interpret motion from a reference video and recover joint rotations to animate the rigged mesh according to the video. Animating and rendering the mesh can be an alternative way to provide feedback for rig verification. We demonstrate examples of this animation and verification in the Supplementary Video.

Modeling of atypical morphologies. Our approach can also find application in representing atypical morphologies. We illustrate this with an example of a polydactyl hand in Figure 7. In this case, we use TRIPO AI's image-to-3D generation model to create a mesh from the input image and then use our pipeline to construct a skeleton adapted to the hand's structure. This exploratory representation supports structural visualization and does not represent reconstruction of clinically verified bone anatomy.

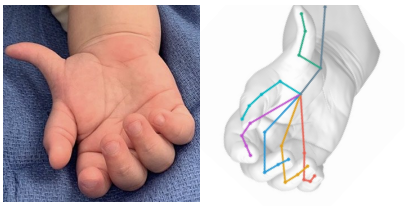


Figure 7: **Atypical morphology.** Skeleton visualization of a polydactyl hand.

6 Conclusion and Future Work

We presented ReRig-Agent, an agent-based approach that treats rigging as an iterative process of perception, construction, and verification. The method integrates visual and geometric evidence to construct skeletons and skinning weights, then uses articulated renderings to inspect the resulting rig and guide further refinement. Experiments on structurally diverse objects demonstrate improved mean skeleton and deformation accuracy compared with the evaluated baselines.

Future work can focus on more reliable deformation assessment, better diagnosis of motion and rigging errors, and broader articulation coverage. Improving the consistency of animation from video examples and developing evaluation measures that better capture local visual quality could make agent-based refinement more effective across uncommon object structures.

AI use statement

During the preparation of this paper, generative AI tools assisted with writing and editing, coding, draft review, research discussions, table formatting, and the creation of reference illustrations. AI-generated suggestions served as supporting material, with the authors making the final research and presentation decisions. The authors reviewed and revised the assisted content, checked code and tables against the implementation and experimental records, and assessed illustrations before inclusion. We take full responsibility for the final text, code, figures, results, and claims.

REFERENCES

Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM Trans. Graph.*, 27(3):44:1–44:10, 2008. doi: 10.1145/1360612.1360643.

Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, jul 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276467. URL https://doi.org.

Zedong Chu, Feng Xiong, Meiduo Liu, Jinzhi Zhang, Mingqi Shao, Zhaoxu Sun, Di Wang, and Mu Xu. Humanrig: Learning automatic rigging for humanoid character in a large scale dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 304–313. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00037. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chu_HumanRig_Learning_Automatic_Rigging_for_Humanoid_Character_in_a_Large_CVPR_2025_paper.html.

Yufan Deng, Yuhao Zhang, Chen Geng, Shangzhe Wu, and Jiajun Wu. Anymate: A dataset and baselines for learning 3d object rigging. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715402. doi: 10.1145/3721238.3730743. URL https://doi.org/10.1145/3721238.3730743.

Olivier Dionne and Martin de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 173–180, 2013. doi: 10.1145/2485895.2485919.

Jingfeng Guo, Jian Liu, Jinnan Chen, Shiwei Mao, Changrong Hu, Puhua Jiang, Junlin Yu, Jing Xu, Qi Liu, Lixin Xu, Zhuo Chen, and Chunchao Guo. Auto-Connect: Connectivity-preserving RigFormer with direct preference optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2913. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7dff60c3db3c0d2f3acf92f13b1b2472-Abstract-Conference.html.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM agent for synthesizing 3D scene as blender code. *arXiv preprint arXiv:2403.01248*, 2024. URL https://arxiv.org/abs/2403.01248.

Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. L_1-medial skeleton of point cloud. *ACM Trans. Graph.*, 32(4):65:1–65:8, 2013. doi: 10.1145/2461912.2461913. URL https://doi.org/10.1145/2461912.2461913.

Doug L James and Christopher D Twigg. Skinning mesh animations. *ACM Transactions on Graphics (TOG)*, 24(3):399–407, 2005.

Binh H. Le and Zhigang Deng. Smooth skinning decomposition with rigid bones. *ACM Transactions on Graphics*, 31(6):199:1–199:10, nov 2012. doi: 10.1145/2366145.2366218.

Binh Huy Le and Zhigang Deng. Robust and accurate skeletal rigging from mesh sequences. *ACM Trans. Graph.*, 33(4):84:1–84:10, 2014. doi: 10.1145/2601097.2601161. URL https://doi.org/10.1145/2601097.2601161.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-Anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.13882.

T. C. Lee, R. L. Kashyap, and C. N. Chu. Building skeleton models via 3-d medial surface/axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994. doi: 10.1006/cgip.1994.1042.

Ruining Li, Yuxin Yao, Matt Zhou, Chuanxia Zheng, Christian Rupprecht, Joan Lasenby, Shangzhe Wu, and Andrea Vedaldi. Instruct-Particulate: Scaling feed-forward 3D object articulation with kinematic control. *arXiv preprint arXiv:2606.14699*, 2026. URL https://arxiv.org/abs/2606.14699.

Isabella Liu, Zhan Xu, Wang Yifan, Hao Tan, Zexiang Xu, Xiaolong Wang, Hao Su, and Zifan Shi. Riganything: Template-free autoregressive rigging for diverse 3d assets. *ACM Trans. Graph.*, 44(4), July 2025. ISSN 0730-0301. doi: 10.1145/3731149. URL https://doi.org/10.1145/3731149.

Pengyu Long, Weirui Wang, Qingcheng Zhao, Xiaoyang Guo, Xiaoyu Pan, Qixuan Zhang, Jiaqing Zhou, Tianlei Hu, Wei Yang, Lan Xu, and Jingyi Yu. Act: A unified framework for rigging and animating characters with arbitrary topologies. *ACM Trans. Graph.*, 45(4), July 2026. ISSN 0730-0301. doi: 10.1145/3811392. URL https://doi.org/10.1145/3811392.

Sining Lu, Guan Chen, Nam Anh Dinh, Itai Lang, Ari Holtzman, and Rana Hanocka. LL3M: Large language 3D modelers. *arXiv preprint arXiv:2508.08228*, 2025. URL https://arxiv.org/abs/2508.08228.

OpenAI. GPT Image 2.5 Introduction Blog. https://openai.com/index/introducing-chatgpt-images-2-5/.

Jia peng Zhang, Cheng-Feng Pu, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. Skin tokens: A learned compact representation for unified autoregressive rigging, 2026. URL https://arxiv.org/abs/2602.04805.

Cheng-Feng Pu, Jia-Peng Zhang, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. Topocap: Learning topology-agnostic motion priors for monocular video-to-animation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH Conference Papers '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725548. doi: 10.1145/3799902.3811159. URL https://doi.org/10.1145/3799902.3811159.

Scott Schaefer and Can Yuksel. Example-based skeleton extraction. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pp. 153–162, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-46-3. URL https://acm.org.

Chaoyue Song, Xiu Li, Fan Yang, Zhongcong Xu, Jiacheng Wei, Fayao Liu, Jiashi Feng, Guosheng Lin, and Jianfeng Zhang. Puppeteer: Rig and animate your 3D models. *arXiv preprint arXiv:2508.10898*, 2025a. doi: 10.48550/arXiv.2508.10898. URL https://arxiv.org/abs/2508.10898.

Chaoyue Song, Jianfeng Zhang, Xiu Li, Fan Yang, Yiwen Chen, Zhongcong Xu, Jun Hao Liew, Xiaoyang Guo, Fayao Liu, Jiashi Feng, and Guosheng Lin. Magicarticulate: Make your 3d models articulation-ready. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 15998–16007, June 2025b.

Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. Mean curvature skeletons. *Comput. Graph. Forum*, 31(5):1735–1744, 2012. doi: 10.1111/J.1467-8659.2012.03178.X. URL https://doi.org/10.1111/j.1467-8659.2012.03178.x.

Zijie Wu, Lixin Xu, Puhua Jiang, Sicong Liu, Chunchao Guo, and Xiang Bai. R-dmesh: Video-guided 3d animation via rectified dynamic mesh flow: R-dmesh. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH Conference Papers '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725548. doi: 10.1145/3799902.3811135. URL https://doi.org/10.1145/3799902.3811135.

Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: neural rigging for articulated characters. *ACM Trans. Graph.*, 39(4):58, 2020. doi: 10.1145/3386569.3392379. URL https://doi.org/10.1145/3386569.3392379.

Zhan Xu, Yang Zhou, Li Yi, and Evangelos Kalogerakis. Morig: Motion-aware rigging of character meshes from point clouds. In Soon Ki Jung, Jehee Lee, and Adam W. Bargteil (eds.), *SIGGRAPH Asia 2022 Conference Papers, SA 2022, Daegu, Republic of Korea, December 6-9, 2022*, pp. 1:1–1:9. ACM, 2022. doi: 10.1145/3550469.3555390. URL https://doi.org/10.1145/3550469.3555390.

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. SceneWeaver: All-in-one 3D scene synthesis with an extensible and self-reflective agent. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2509.20414.

Hao Zhang, Jiahao Luo, Bohui Wan, Yizhou Zhao, Zongrui Li, Michael Vasilkovsky, Chaoyang Wang, Jian Wang, Narendra Ahuja, and Bing Zhou. Rigmo: Unifying rig and motion learning for generative animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25438–25449, June 2026a.

Hongxin Zhang, Chunru Lin, Junyan Li, Zhou Xian, Tsun-Hsuan Wang, and Chuang Gan. GS-Agent: Creating 4D physical worlds with generative simulation. *arXiv preprint arXiv:2607.21522*, 2026b. URL https://arxiv.org/abs/2607.21522.

Jia-Peng Zhang, Cheng-Feng Pu, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. One model to rig them all: Diverse skeleton rigging with unirig. *ACM Trans. Graph.*, 44(4), July 2025a. ISSN 0730-0301. doi: 10.1145/3730930. URL https://doi.org/10.1145/3730930.

Shuyuan Zhang, Chenhan Jiang, Zuoou Li, and Jiankang Deng. ShapeCraft: LLM agents for structured, textured and interactive 3D modeling. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2510.17603.

Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3D asset generation. *arXiv preprint arXiv:2605.15187*, 2026. URL https://arxiv.org/abs/2605.15187.