One Model to Rig Them All: Diverse Skeleton Rigging with UniRig

Presenter: **Ji Yang**

V&L Lab @ ECE

January 29, 2026

Roadmap

Paper published in SIGGRAPH(TOG) 2025.

1. Motivation and problem definition
2. Prior work: template-based vs template-free rigging
3. UniRig overview (two-stage design)
4. Stage I: Autoregressive skeleton tree generation
5. Stage II: Skinning weights via bone–point cross-attention
6. Training tricks: skeletal equivalence + physical simulation supervision
7. Experiments and qualitative analysis
8. Strengths, limitations, and research opportunities

Why Auto-Rigging Still Matters

- 3D content creation is accelerating (AI-generated assets + traditional pipelines).
- Manual rigging remains a bottleneck: time-consuming, expertise-intensive.
- Industry pipelines still heavily rely on **skeleton + skinning + retargeting**.

Paper Fig.2: Examples from Rig-XL, demonstrating well-defined skeleton structures.

What is Skeletal Rigging? (Formalization)

Given a mesh (or surface):

$$\mathcal{M} = \{\mathbf{V} \in \mathbb{R}^{|\mathcal{V}|\times 3},\ \mathbf{F}\},$$

predict:

- **Skeleton joints/bones:** joint positions $\mathbf{J} \in \mathbb{R}^{J\times 3}$ and parent indices $\mathbf{P} \in \mathbb{N}^{J-1}$ forming a **tree**.
- **Skinning weights:** $\mathbf{W} \in \mathbb{R}^{N\times J}$ (per-vertex influence).
- (Optional) **Bone attributes** (e.g., stiffness/gravity parameters) $\mathbf{A} \in \mathbb{R}^{J\times B}$.

UniRig explicitly frames the skeleton as a hierarchical tree and targets both skeleton + weights in one pipeline [Zhang et al. 2025].

Key Challenges

(1) Skeleton is a **tree**, not a point set

Connectivity constraints (acyclic, rooted, hierarchical) are hard to enforce in regression-based methods.

(2) Shape/topology diversity

Humans, quadrupeds, insects, birds, fictional characters, even semi-static objects.

(3) Skinning is a **global** bone–vertex interaction

Many-to-many relationships: $N \times J$ can be huge (tens of thousands vertices; hundreds bones).

Prior Work Taxonomy

- **Template-based**: strong accuracy but limited topology (e.g., SMPL-like, Mixamo-like).
- **Template-free**: more general but often unstable skeleton topology (e.g., RigNet [Xu et al. 2020]).
- **Skeleton-free deformation**: bypass skeleton, less compatible with standard pipelines.

UniRig idea

A **unified** framework targeting **diverse** skeleton topologies, while generating **topologically valid** trees and accurate skinning.

Why Template-free Methods Struggle (Intuition)

Common pipeline:

1. Predict joints (heatmap / regression) ⇒ noisy set of points
2. Post-process connectivity (Min. Spanning Tree / heuristics) ⇒ topology errors

Failure mode

Connectivity is inferred indirectly, so it is easy to create implausible or unstable skeleton structures.

UniRig idea

Generate the tree directly using an autoregressive model with a tree-aware tokenization.

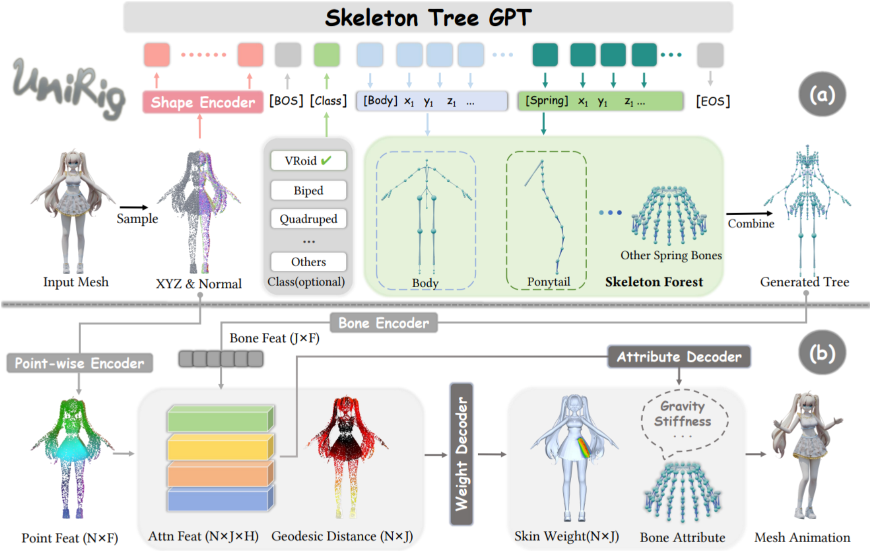
UniRig: Two-Stage Pipeline

- Stage I: **Autoregressive skeleton tree prediction** from mesh/point cloud
- Stage II: **Skinning weight prediction** conditioned on predicted skeleton via bone–point cross-attention

UniRig idea

Combining an autoregressive model for skeleton prediction with bone–point cross-attention for weights.

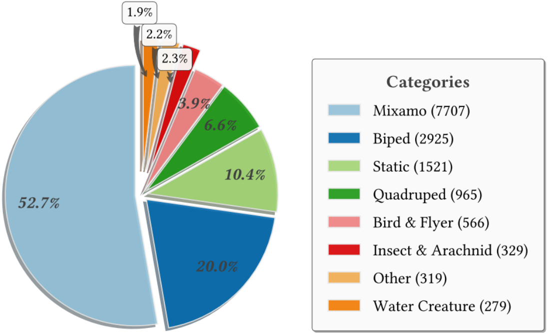
UniRig: Two-Stage Pipeline



Paper Fig.5: Overview of the UniRig framework.

Data: VRoid + Rig-XL

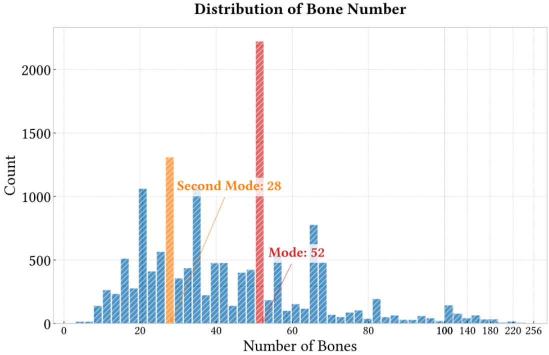
- **VRoid:** 2,061 anime-style humanoid models (VRM format; includes spring bones).
- **Rig-XL:** 14,611 rigged models curated from Objaverse-XL subset [Deitke et al. 2024], spanning 8 categories.



Paper Fig.3: Category distribution of Rig-XL. The percentages indicate the proportion of models belonging to each category.

Data: VRoid + Rig-XL

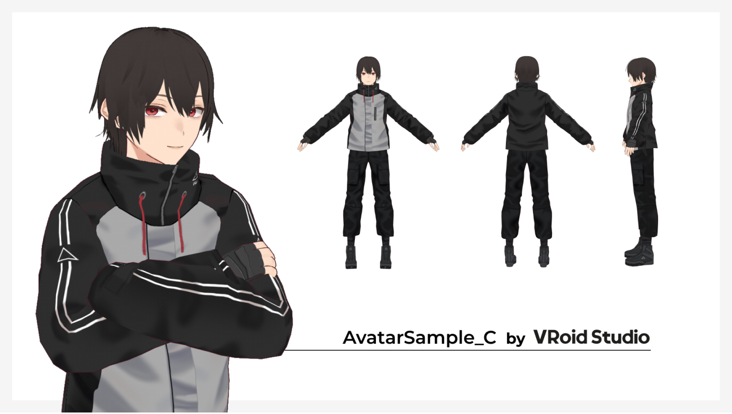
- **VRoid:** 2,061 anime-style humanoid models (VRM format; includes spring bones).
- **Rig-XL:** 14,611 rigged models curated from Objaverse-XL subset [Deitke et al. 2024], spanning 8 categories.



Paper Fig.4: Distribution of bone numbers in Rig-XL. The histogram shows the frequency of different bone counts across all models in the dataset.

Data: VRoid + Rig-XL

- **VRoid:** 2,061 anime-style humanoid models (VRM format; includes spring bones).
- **Rig-XL:** 14,611 rigged models curated from Objaverse-XL subset [Deitke et al. 2024], spanning 8 categories.



A sample model from VRoid.

Rig-XL Curation (Why it matters)

Rig-XL curation steps (high-level):

1. Skeleton-based filtering (bone count range, single connected tree).
2. Automated categorization using rendered views + VLM captions.
3. Manual verification & refinement to fix common skeleton errors.

This pipeline is explicitly described to address missing skeleton/weights, multi-object scenes, and category bias in source data [Zhang et al. 2025].

Stage I: Input Representation

UniRig converts a mesh into a point cloud:

$$\mathbf{X} \in \mathbb{R}^{N \times 3}, \quad \mathbf{N} \in \mathbb{R}^{N \times 3}$$

where $N = 65536$ surface points sampled and normalized to $[-1, 1]^3$.

- A geometric encoder $E_G : (\mathbf{X}, \mathbf{N}) \mapsto \mathbf{F}_G$ produces a conditioning embedding.
- An OPT-style decoder-only transformer [Zhang et al. 2022] generates a token sequence representing the skeleton tree.

Autoregressive Modeling: Next-Token Prediction

Let the tokenized skeleton be a sequence

$$\mathbf{S} = (s_1, s_2, \ldots, s_T).$$

Training uses next-token prediction:

$$\mathcal{L}_{\mathrm{NTP}} = -\sum_{t=1}^{T} \log P\left(s_t \mid s_{<t}, \mathbf{F}_G\right).$$

- Causal generation models hierarchical dependencies.
- Conditioning on $\mathbf{F}_G$ ties structure to geometry [Zhang et al. 2025].

Core Difficulty: How to Tokenize a Tree?

A skeleton is a rooted tree with:

- spatial coordinates of joints/bones
- parent–child relations
- special bone types (templates, spring bones, etc.)

Naïve idea

Serialize in DFS/BFS and repeatedly include parent coordinates for each child.

Problem

Redundant tokens, harder constraint enforcement, repetitive sequences during inference.

Skeleton Tree Tokenization: Coordinate Discretization

UniRig discretizes coordinates in $[-1, 1]$ into $D = 256$ bins:

$$M : x \in [-1, 1] \mapsto d = \left\lfloor \frac{x+1}{2} D \right\rfloor \in \mathbb{Z}_D, \qquad M^{-1} : d \mapsto x = \frac{2d}{D} - 1.$$

Each joint/bone coordinate becomes discrete tokens (d_x, d_y, d_z).

Tokenization: Structural Tokens and Compression

UniRig adds:

- **Class token** $\langle C \rangle$ (e.g., VRoid / Mixamo / Quadruped)
- **Type identifiers** $\langle$spring_bone$\rangle$, $\langle$mixamo:body$\rangle$, ...
- **Branch token** $\langle$branch$\rangle$ to encode a forest of chains (DFS extraction)

Example of Tokenization

$<\textbf{bos}>\ <\textbf{VRoid}>\ <\textbf{mixamo:body}>\ dx_1\ dy_1\ dz_1 \ldots dx_{22}\ dy_{22}\ dz_{22}$
$<\textbf{mixamo:hand}>\ dx_{23}\ dy_{23}\ dz_{23} \ldots dx_{52}\ dy_{52}\ dz_{52} \ldots$
$<\textbf{spring\_bone}>\ dx_s\ dy_s\ dz_s \ldots dx_t\ dy_t\ dz_t \ldots\ <\textbf{eos}>$

Algorithmic View (Short Version)

1. Match template bones (e.g., Mixamo) and emit a template token $+$ coordinates.
2. Remove template bones $\Rightarrow$ remaining forest.
3. DFS to extract bone chains; sort children by (z, y, x); emit $\langle$branch$\rangle$ markers.
4. De-tokenization merges joints whose decoded positions are within a distance threshold.

Please refer to the original paper for the full algorithmic details.

Token Savings (Quantitative)

Average token cost reduction:

- VRoid: 667.27 $\rightarrow$ 483.95 (27.47% reduction)
- Rig-XL: 266.28 $\rightarrow$ 187.15 (29.72% reduction)

Interpretation

Shorter sequences $\Rightarrow$ less memory, faster training/inference, and fewer repetitive-generation artifacts [Zhang et al. 2025].

Why Autoregressive Helps Topology

Key conceptual shift

Connectivity is not post-processed. It is **generated as part of the sequence**.

- Conditional generation allows long-range constraints (global tree consistency).
- Structured tokens act like a "grammar" for valid skeletons.
- Template-aware tokens encode priors for retargeting and special bones.

Problem: Predict Skinning Weights at Scale

Goal: weight matrix

$$\mathbf{W} \in \mathbb{R}^{N \times J}$$

where N can be 10^4–10^5, J up to hundreds.
Also predict bone attributes:

$$\mathbf{A} \in \mathbb{R}^{J \times B}$$

(e.g., stiffness, gravity coefficients for spring bones) [Zhang et al. 2025].

Encoders

UniRig uses two feature encoders:

- **Bone encoder** E_B (MLP + positional encoding):

$$(\mathbf{J}^P, \mathbf{J}) \in \mathbb{R}^{J \times 6} \mapsto \mathbf{F}_B \in \mathbb{R}^{J \times F}.$$

- **Point encoder** E_P (pretrained Point Transformer V3 [Wu et al. 2024] from SAMPart3D [Yang et al. 2024]):

$$\mathbf{X} \in \mathbb{R}^{N \times 3} \mapsto \mathbf{F}_P \in \mathbb{R}^{N \times F}.$$

The paper emphasizes using a strong pretrained point backbone for fine-grained geometry [Zhang et al. 2025].

Bone–Point Cross Attention (Weights)

Project features into queries/keys/values:

$$\mathbf{Q}_W = \mathbf{F}_P \mathbf{W}_Q, \quad \mathbf{K}_W = \mathbf{F}_B \mathbf{W}_K, \quad \mathbf{V}_W = \mathbf{F}_B \mathbf{W}_V.$$

Cross-attention weights (with H heads):

$$\mathbf{F}_W = \mathrm{softmax}\left(\frac{\mathbf{Q}_W \mathbf{K}_W^\top}{\sqrt{F}}\right) \in \mathbb{R}^{N \times J \times H}.$$

Interpretation

Each vertex “queries” which bones explain it; bones provide keys/values [Zhang et al. 2025].

Augment with Geodesic Distance + Decode Weights

Let $\mathbf{D} \in \mathbb{R}^{N \times J}$ be voxel geodesic distances between vertices and bones (precomputed). Then:

$$\mathbf{W} = \mathrm{softmax}\left(E_W\left(\mathrm{concat}(\mathbf{F}_W, \mathbf{D})\right)\right).$$

- Geodesic distance provides topology-aware proximity (better than Euclidean for articulated surfaces).
- Final softmax ensures per-vertex weight normalization.

This is explicitly described as concatenating **D** with attention features then MLP + softmax [Zhang et al. 2025].

Reverse Attention for Bone Attributes

To predict attributes $\mathbf{A}$, swap roles:

$$\mathbf{A} = E_A\big(\text{cross\_attn}(\mathbf{F}_B, \mathbf{F}_P)\big).$$

- Bones query points to aggregate relevant geometric context.
- Needed for spring bone simulation parameters (stiffness, gravity, etc.).

Supervision: Direct Losses

UniRig uses:

- KL divergence for skinning distributions:

$$\mathcal{L}_W = \mathrm{KL}\big(\mathbf{W} \,\|\, \mathbf{W}_{\text{pred}}\big)$$

- ℓ_2 loss for attributes:

$$\mathcal{L}_A = \|\mathbf{A} - \mathbf{A}_{\text{pred}}\|_2^2$$

Combined:

$$\lambda_W \mathcal{L}_W + \lambda_A \mathcal{L}_A.$$

The paper states KL for weights and L2 for attributes [Zhang et al. 2025].

Training Issue: Bone Imbalance

Naïvely sampling points uniformly biases optimization:

- Large bones (hips/torso) get many vertices $\Rightarrow$ dominate gradients.
- Small / sparse regions (hair/fingers) under-trained.

UniRig solution

Skeletal equivalence training: encourage each bone to contribute equally [Zhang et al. 2025].

Skeletal Equivalence: Two Mechanisms

1. **Random bone freezing** (probability p): frozen bones use GT weights, no gradients.
2. **Bone-centric loss normalization**: average loss per bone rather than per vertex (prevent domination).

A representative normalized form (conceptual):

$$\frac{1}{J}\sum_{i=1}^{J}\frac{1}{S_i}\sum_{k=1}^{N}\mathbf{1}[W_{k,i}>0]\,\ell_k, \quad S_i=\sum_{k=1}^{N}\mathbf{1}[W_{k,i}>0].$$

Normalize per bone and then average over bones [Zhang et al. 2025].

Why Direct Weight Loss is Not Enough

Multiple weight solutions can yield similar deformations under simple motions.

Problem

Direct supervision may not guarantee visually realistic motion (especially with spring bones).

UniRig solution

Add **indirect supervision** via differentiable physical simulation (Verlet-style spring bone dynamics) [Zhang et al. 2025].

Indirect Supervision via Physical Simulation (High-Level)

Sample a short motion sequence $\mathcal{M}$ (length $T = 3$) and apply it to:

- predicted parameters $(\mathbf{W}_{\text{pred}}, \mathbf{A}_{\text{pred}})$
- ground truth $(\mathbf{W}, \mathbf{A})$

Obtain simulated vertex sequences:

$$\mathbf{X}_{\text{pred}}^{\mathcal{M}} \quad \text{and} \quad \mathbf{X}^{\mathcal{M}}.$$

Use reconstruction loss:

$$\mathcal{L}_X = \sum_{t=1}^{T} \left\| \mathbf{X}_t^{\mathcal{M}} - \mathbf{X}_{\text{pred},t}^{\mathcal{M}} \right\|_2^2.$$

Final training objective:

$$\lambda_W \mathcal{L}_W + \lambda_A \mathcal{L}_A + \lambda_X \mathcal{L}_X.$$

This exact 3-term objective is described in the paper [Zhang et al. 2025].

Evaluation Setup

Two evaluation axes:

1. **Skeleton accuracy** (geometry + topology consistency)
2. **Skinning quality / animation robustness** (weights that drive realistic motion)

Datasets:

- VRoid (detailed humanoids with spring bones)
- Mixamo-like skeletons (template-ish)
- Rig-XL (diverse objects)

Skeleton Metrics: Chamfer-style Distances

The paper uses three Chamfer-based measures (conceptually):

- J2J: predicted joints vs GT joints
- J2B: predicted joints to closest points on GT bones
- B2B: predicted bones vs GT bones

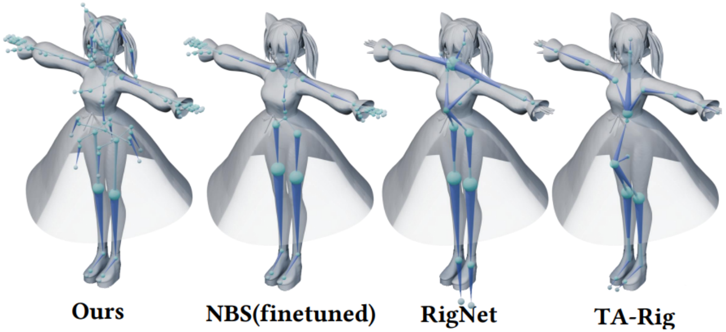
Method	Mixamo	VRoid	Mixamo⋆	VRoid⋆	Rig-XL⋆
Ours	**0.0101**	**0.0092**	**0.0103**	**0.0101**	**0.0549**
RigNet† [Xu et al. 2020]	0.1022	0.2405	0.2171	0.2484	0.2388
NBS [Li et al. 2021]	0.0338	0.0205	0.0429	0.0214	N/A
TA-Rig† [Ma and Zhang 2023]	0.1007	0.0886	0.1093	0.0934	0.2175

Table: Quantitative comparison of Joint-to-Joint Chamfer Distance (J2J). ⋆ means the evaluation dataset is under the data augmentation of random rotation, scale, and applying random motion. † indicates the model cannot be finetuned due to unavailable code base.

For J2B and B2B results, refer to the supplementary.

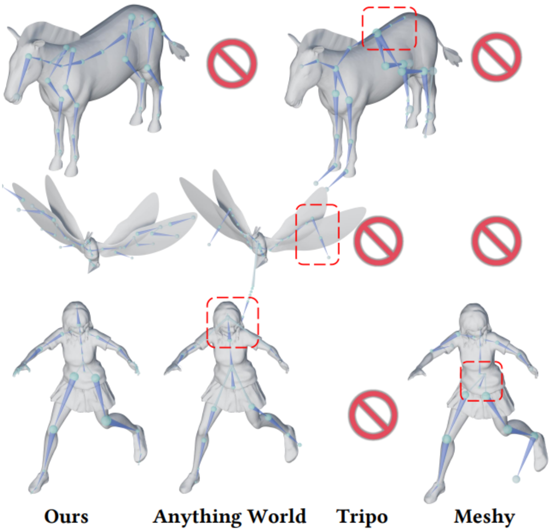
Qualitative Skeleton Comparison

- Compare against RigNet [Xu et al. 2020], NBS [Li et al. 2021], TA-Rig [Ma and Zhang 2023], and commercial tools.
- Common observed improvements: more complete skeletons, fewer topology failures.



Paper Fig.7: Comparison of predicted skeletons between NBS (fine-tuned), RigNet, and TA-Rig on the VRoid dataset.

Qualitative Skeleton Comparison



Paper Fig.8: Qualitative comparison of predicted skeletons against commercial tools.

Skinning Weight Accuracy

Method	Mixamo	VRoid	Mixamo⋆	VRoid⋆	Rig-XL⋆
Ours	**0.0055**	**0.0028**	**0.0059**	**0.0038**	0.0329
RigNet† [Xu et al. 2020]	0.04540	0.04893	0.05367	0.06146	N/A
NBS [Li et al. 2021]	0.07898	0.02721	0.08211	0.03339	N/A

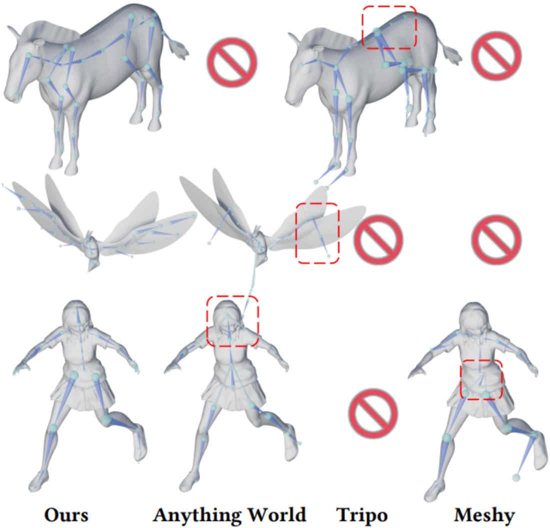
Table: Comparison of skinning weight prediction accuracy using per-vertex L1 loss between predicted and ground-truth skinning weights. ⋆ means the evaluation dataset is under the data augmentation of random rotation, scale, and applying random motion. † indicates the model cannot be finetuned due to unavailable code base.

Key idea

Cross-attention + geodesic distance produces more consistent bone influence patterns [Zhang et al. 2025].

Animation Robustness (Motion Reconstruction)

Evaluate robustness by applying many animation sequences.



Paper Fig.9: Qualitative comparison of mesh deformation under motion.

Takeaways

1. **Tree-as-sequence tokenization** enables direct topology generation (no MST post-processing).
2. **Bone–point cross-attention** scales skinning to large $N \times J$ while injecting structure.
3. **Indirect physical supervision** pushes weights/attributes toward motion realism, not just per-vertex loss.
4. **Rig-XL and VRoid dataset** increases diversity and drives generalization.

Strengths

- Treats skeleton as a structured object and gives the model a “grammar”.
- Template-aware design supports practical retargeting scenarios.
- Uses strong pretrained point features (reduces data hunger for geometry).
- Adds motion-level supervision that aligns with animation quality.

Limitations / Open Questions

(1) Tree assumption

Skeleton is modeled as a rooted tree. How well does it handle:

- mechanical rigs with loops / constraints?
- multi-rooted rigs / accessories?

(2) Dependency on dataset curation

Rig-XL relies on VLM-based categorization and manual refinement, which may encode biases or heuristics [Zhang et al. 2025].

(3) Computational components

Geodesic distance and autoregressive decoding can be heavy at scale; what is the true production cost?

Summary

What UniRig contributes

A unified auto-rigging framework that:

- generates **topologically valid skeleton trees** via autoregressive token generation
- predicts **high-quality skinning weights** via bone–point cross-attention + geodesic priors
- improves motion realism through **physics-based indirect supervision**
- is trained on a new large, diverse dataset (Rig-XL)

References

- Deitke, Matt et al. (2024). "Objaverse-XL: A Universe of 10M+ 3D Objects". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36.
- Li, Peizhuo et al. (2021). "Learning Skeletal Articulations with Neural Blend Shapes". In: *ACM Transactions on Graphics (TOG)* 40.4, pp. 1–15.
- Ma, Jing and Dongliang Zhang (2023). "TARig: Adaptive Template-aware Neural Rigging for Humanoid Characters". In: *Computers & Graphics* 114, pp. 158–167.
- Wu, Xiaoyang et al. (2024). "Point Transformer V3: Simpler Faster Stronger". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851.
- Xu, Zhan et al. (2020). "RigNet: Neural Rigging for Articulated Characters". In: *ACM Transactions on Graphics (TOG)* 39.4, pp. 1–14.
- Yang, Yunhan et al. (2024). "SAMPart3D: Segment Any Part in 3D Objects". In: *arXiv preprint arXiv:2411.07184*.
- Zhang, Jia-Peng et al. (2025). "One Model to Rig Them All: Diverse Skeleton Rigging with UniRig". In: *ACM Transactions on Graphics (TOG)* 44.4. Special Issue on SIGGRAPH 2025.
- Zhang, Susan et al. (2022). "OPT: Open Pre-trained Transformer Language Models". In: *arXiv preprint arXiv:2205.01068*.