Tencent 腾讯

ICML
International Conference
On Machine Learning

FreeMesh: Boosting Mesh Generation with Coordinates Merge

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

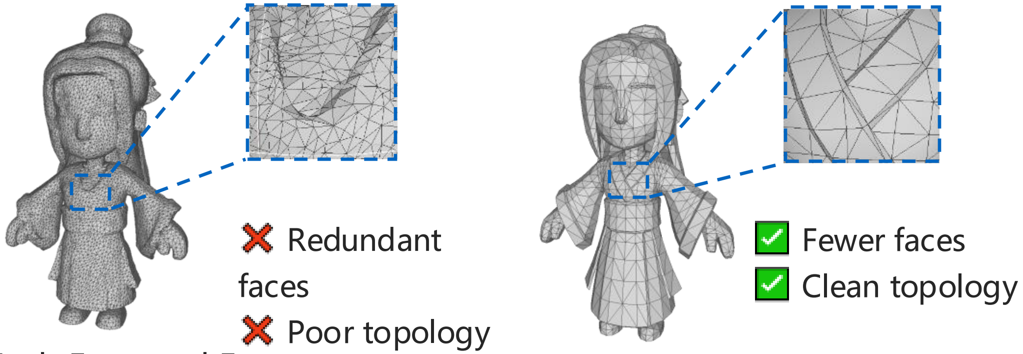
Jian Liu, HKUST&Hunyuan3D

腾讯混元

Native Mesh Generation: Taking 3D Generation from "Viewable" to "Usable"

Current Key Challenges in 3D Generation: The Inability to Produce Artist-like Meshes

While current 3D models achieve geometric precision, mesh topology quality lags artist standards, precluding direct use (e.g., gaming). Issues include:



Mesh Extracted From Implicit Field

Mesh Crafted by Designers

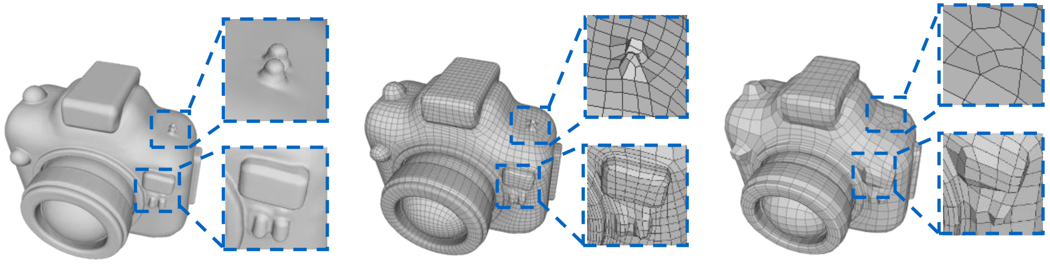
1) **Redundant Faces:** extracted meshes typically contain excessive polygons, which are unsuitable for real-time applications (e.g., games).
2) **Poor Topology:** Chaotic edge flow in extracted meshes blocks animation-ready workflows (e.g., rigging/skinning).

Casual Analysis:

Hunyuan2.0 employs Implicit Field and marching cube for mesh extraction in 3D generation, which only models geometric shapes without learning mesh topology.

Mainstream Solutions and Existing Issues:

Current methods use remesh algorithms for post-processing, improving mesh topology partially. But for low-poly counts in professional pipelines, they cause detail loss and fail on different objects.



3D Generated mesh (1.2 million faces)

Remesh (15k faces)

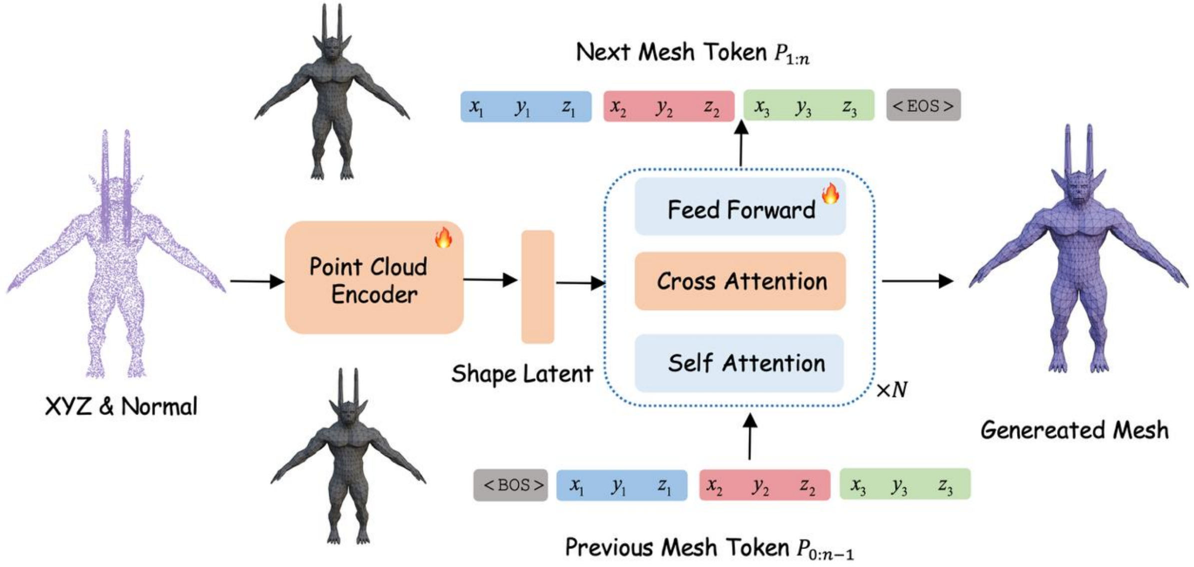
Remesh (5k faces)

Native Mesh Generation: Artistic-Mesh Aligned 3D Generation

Native Mesh Generation via Autoregression — A new paradigm in 3D generation

- Core Idea: Explicitly model vertices and faces, directly learn edge topology from high-quality meshes.

- Formulate mesh generation as next-mesh-token prediction:
  1) Tokenize mesh (triangle soups) into a 1D coordinate sequence.
  2) Model the mesh sequence with GPT-style Transformer.
  3) Decode the sequence back to the final mesh.



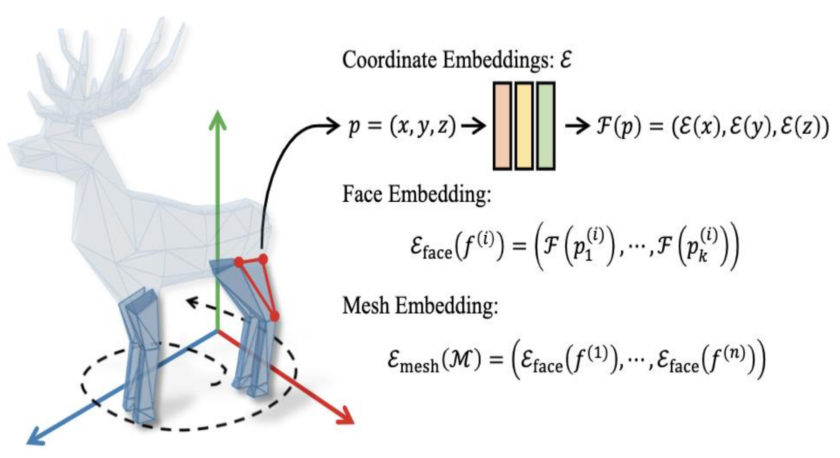
Autoregressive Mesh Generation FrameWork

Examples

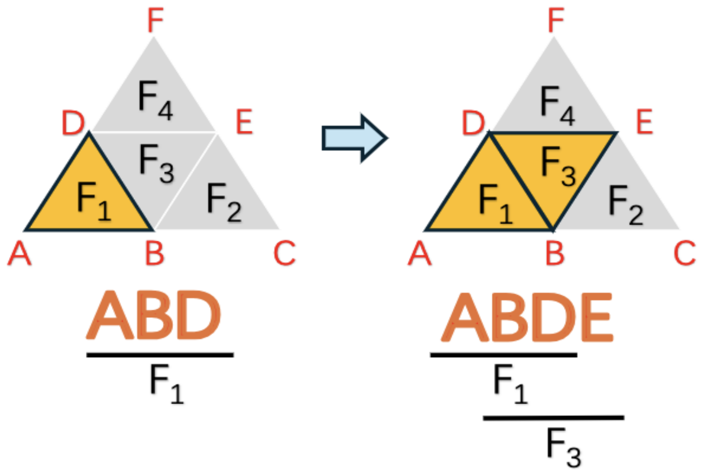
Fewer faces、Clean topology、Part structure

Native Mesh Generation：Current Challenge

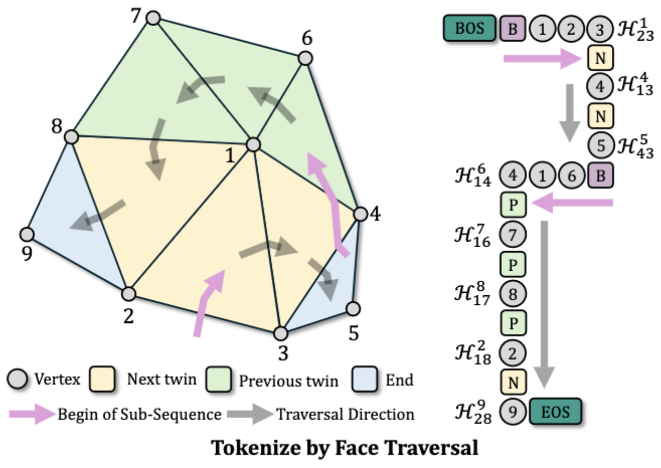
Challenge ：How to develop a better tokenizers and evaluate tokenizers？



MeshXL



MeshAnythingV2



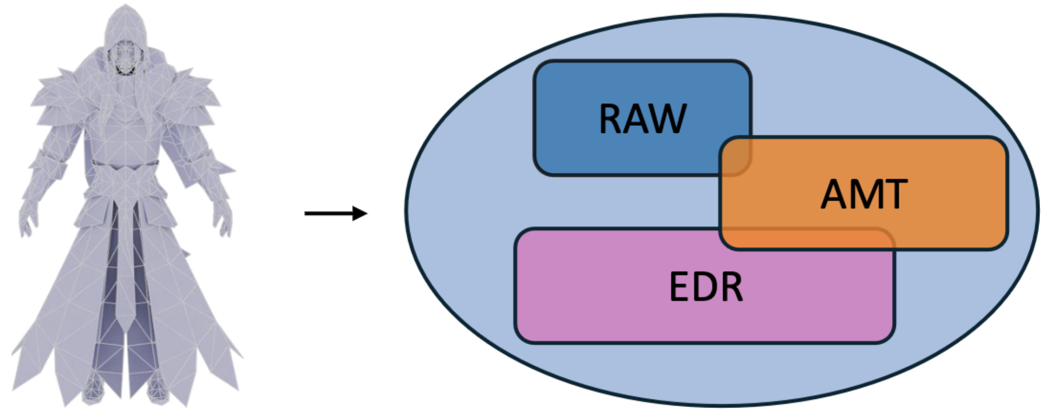
EdgeRunner

Research Background: Existing serialization methods compress meshes by traversing topology and geometry from a facet-based perspective, facing two key limitations:

1. The lack of quantitative evaluation metrics for assessing the effectiveness of such methods.
2. Their compression ratio is capped at approximately 50%. This severely restricts the usable context length for models, making it impossible to effectively generate meshes with complex topology and high polygon counts.

FreeMesh：Rearrange and Merge Coordinates

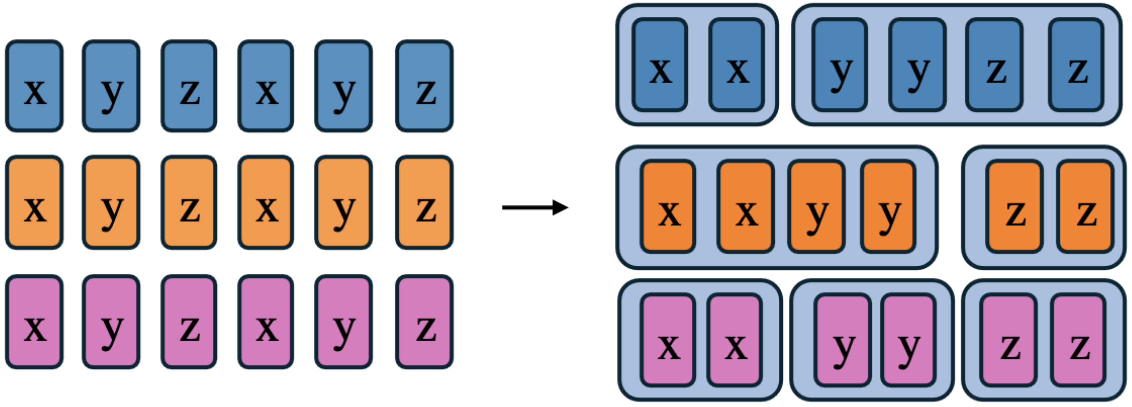
RMC：A New Tokenizer With SOTA Compress Ratio



Input Mesh

Mesh Tokenizers

Geometric Compression(Facet Dimension):
Prior methods (e.g., EdgeRunner) adopt face traversal compress spatial mesh coordinates, achieving ~50% compression rates.



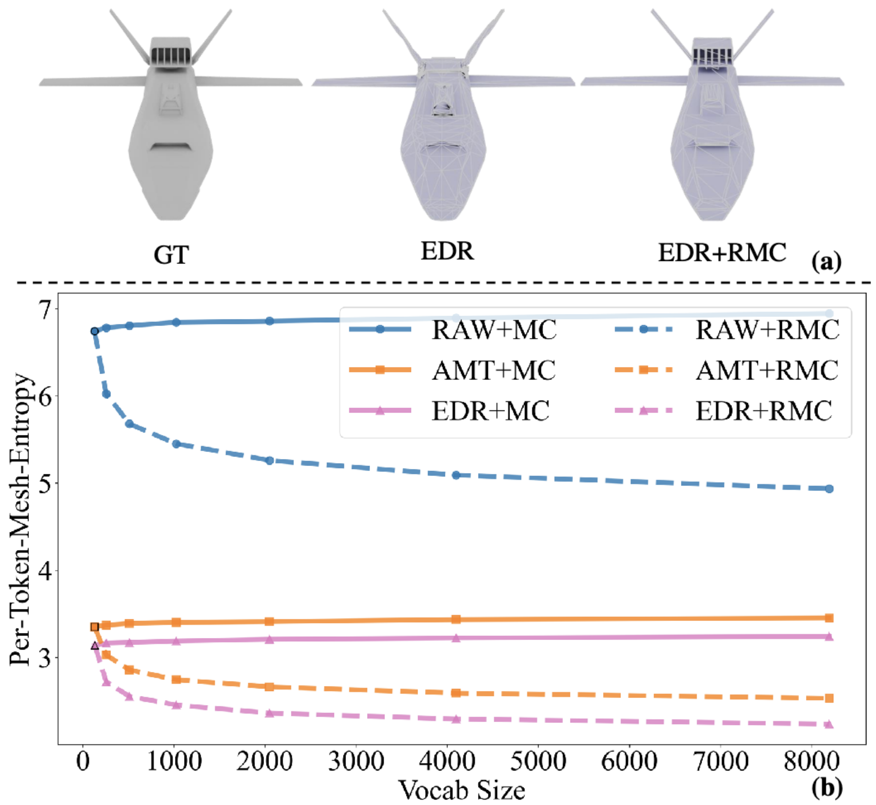
Vanilla Mesh Tokens

Rearrange & Merge

Frequency Compression (Coordinate Dimension):
Our approach merges adjacent coordinate tokens into unified representations, reducing redundancy patterns for additional ~42% compression.

FreeMesh：Per-Token-Mesh-Entropy(PTME)

PTME ：A Metric to Evaluate tokenizers



$$
\begin{aligned}
\mathcal{PTME} &= -\sum_{s\in V_s}\frac{N_s}{|Seq_{V_R}|}\log p_s \\
&= \left(-\sum_{s\in V_s}\frac{N_s}{|Seq_{V_s}|}\log p_s\right)\Big/\left(\frac{|Seq_{V_R}|}{|Seq_{V_s}|}\right) \\
&= \left(-\sum_{s\in V_s} p_s\log p_s\right)\Big/\left(\frac{|Seq_{V_c}|}{C_R\times|Seq_{V_s}|}\right) \\
&= \left(-\sum_{s\in V_s} p_s\log p_s\right)\Big/\left(\frac{\sum_{s\in V_s}N_s l_s}{|Seq_{V_s}|}\right)\times C_R \\
&= \left(-\sum_{s\in V_s} p_s\log p_s\right)\Big/\left(\sum_{s\in V_s} p_s l_s\right)\times C_R \\
&= \frac{\mathcal{H}_s}{l}\times C_R,
\end{aligned}
$$

Naive coordinate merging cannot reduce PTME; only by strategically rearranging coordinates to boost adjacent-pair co-occurrence probability can we substantially lower Per-Token Modeling Entropy.

FreeMesh：Per-Token-Mesh-Entropy(PTME)

Proof: How to decrease PTME?

$$\mathcal{PTME} = \frac{\mathcal{H}}{l} = \frac{-\sum_i p_i \log p_i}{\sum_i p_i l_i} \tag{1}$$

$$\begin{aligned} \tilde{p}_{ab} &= \frac{p_{ab}}{1-p_{ab}}, \\ \tilde{p}_a &= \frac{p_a - p_{ab}}{1-p_{ab}}, \\ \tilde{p}_b &= \frac{p_b - p_{ab}}{1-p_{ab}}, \\ \tilde{p}_i &= \frac{p_i}{1-p_{ab}}, \quad (i \neq a, b) \end{aligned} \tag{2}$$

$$\begin{aligned} \tilde{\mathcal{H}} = -\frac{1}{1-p_{ab}} &\left[p_{ab} \log \frac{p_{ab}}{1-p_{ab}} + \sum_{i=a,b} (p_i - p_{ab}) \log \frac{p_i - p_{ab}}{1-p_{ab}} \right. \\ &\left. + \sum_{i \neq a,b} p_i \log \frac{p_i}{1-p_{ab}} \right] \\ = \frac{1}{1-p_{ab}} &(\mathcal{H} - \mathcal{F}_{ab}) \end{aligned} \tag{3}$$

$$\begin{aligned} \tilde{l} &= \frac{p_{ab}(l_a + l_b) + \sum_{i=a,b}(p_i - p_{ab}) l_i + \sum_{i \neq a,b} p_i l_i}{1-p_{ab}} \\ &= \frac{l}{1-p_{ab}} \end{aligned} \tag{4}$$

$$\frac{\tilde{\mathcal{H}}}{\tilde{l}} - \frac{\mathcal{H}}{l} = -\frac{\mathcal{F}_{ab}}{l} \tag{5}$$

$$\mathcal{F}_{ab} = p_{ab} \log \frac{p_{ab}}{p_a p_b} - (1-p_{ab}) \log(1-p_{ab}) + \sum_{i=a,b} (p_i - p_{ab}) \log \left(1 - \frac{p_{ab}}{p_i}\right) \tag{6}$$

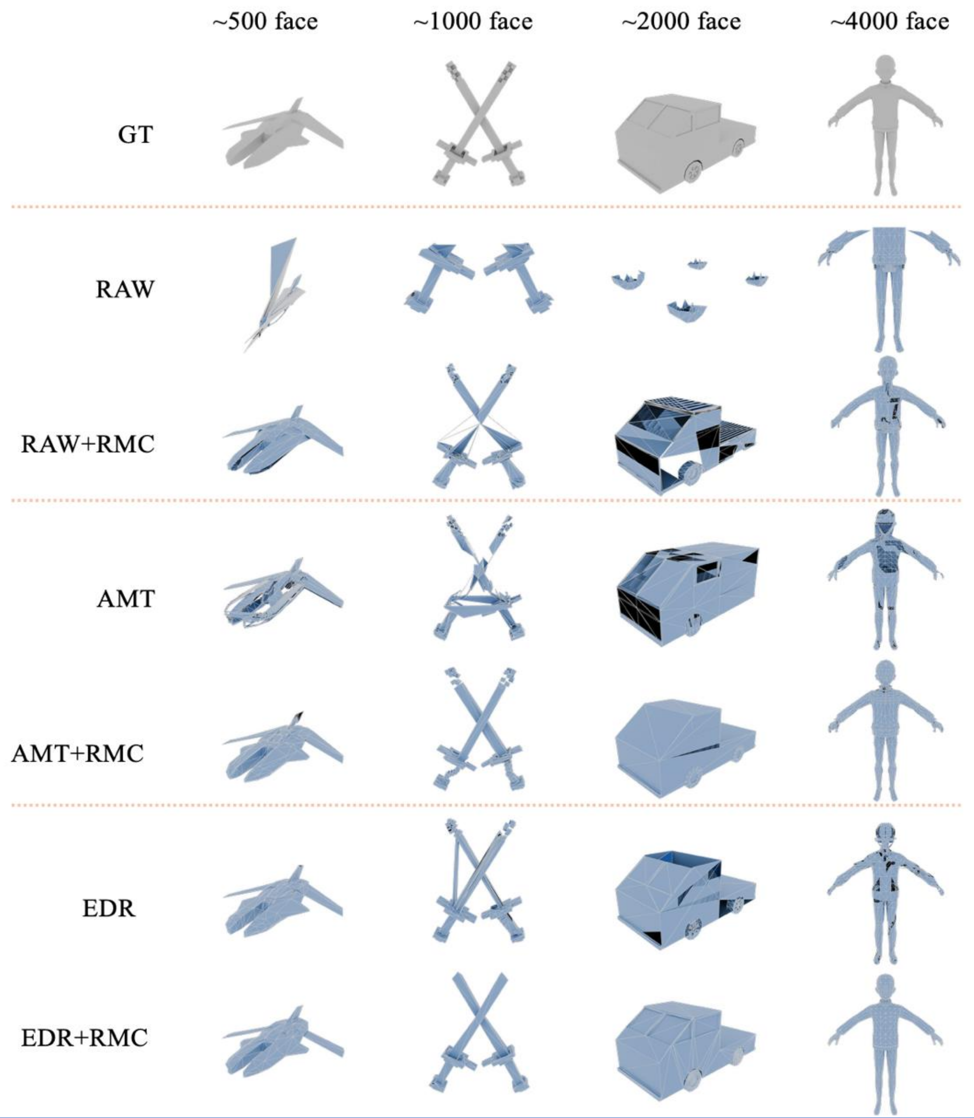
$$\begin{aligned} \ln(1-p_{ab}) &\approx -p_{ab} \\ \ln\left(1 - \frac{p_{ab}}{p_i}\right) &\approx -\frac{p_{ab}}{p_i} \end{aligned} \tag{7}$$

$$\mathcal{F}_{ab} \approx \mathcal{F}^*_{ab} = p_{ab}\left(\ln \frac{p_{ab}}{p_a p_b} - 1\right) \tag{8}$$

Our derivation demonstrates that higher co-occurrence frequency of adjacent coordinate pairs yields greater reduction in Per-Token Modeling Entropy (PTME).

FreeMesh：Experiment Results

Quantitative Results



Qualitative Results

	Compress Ratio	Max Faces	Hausdorff Distance	PTME	Win Rate
RAW	0%	<1k	0.647	6.742	0%
RAW + RMC	54%	<2k	0.543	4.937	8%
AMT	50%	<2k	0.428	3.349	30%
AMT+RMC	25%	~4k	0.325	2.537	62%
EDR	50%	~4k	0.408	3.139	57%
EDR+RMC	**78%**	**~8k**	**0.280**	**2.231**	**95%**

Tencent 腾讯

Thanks for your listening!

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

腾讯混元