RigNet: Neural Rigging for Articulated Characters

Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, Karan Singh

University of Massachusetts Amherst ; University of Toronto

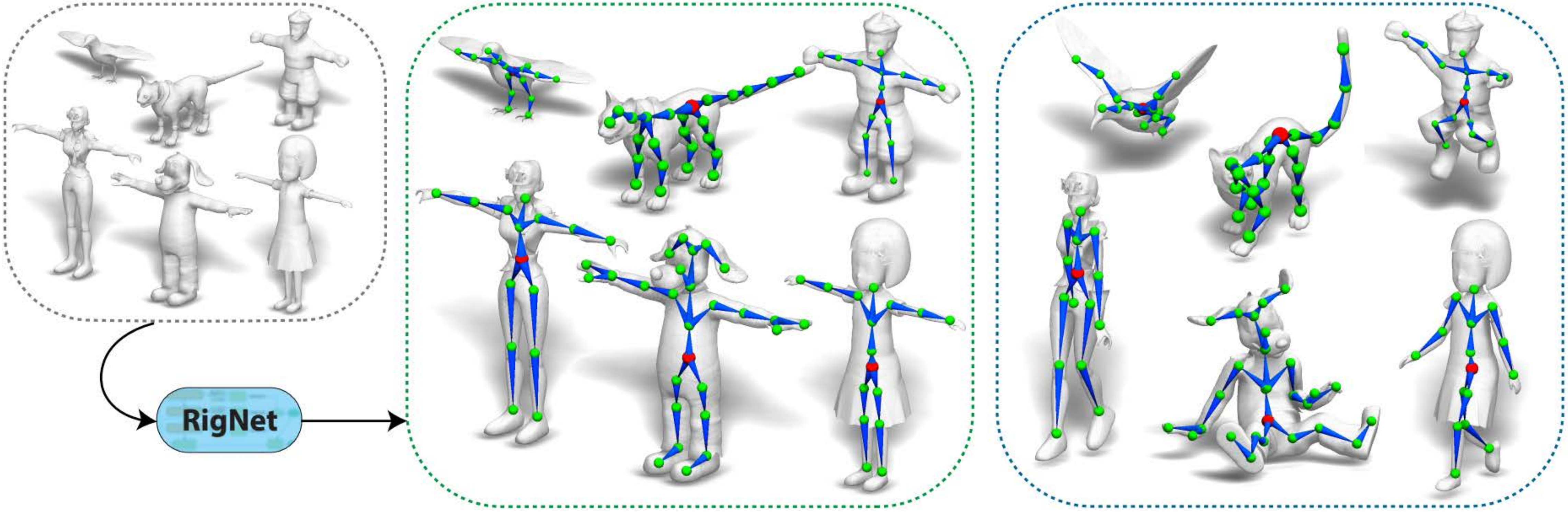
Siggraph 2020

Overall

Our Goal: Drive the mesh

Input: 3D mesh of a character

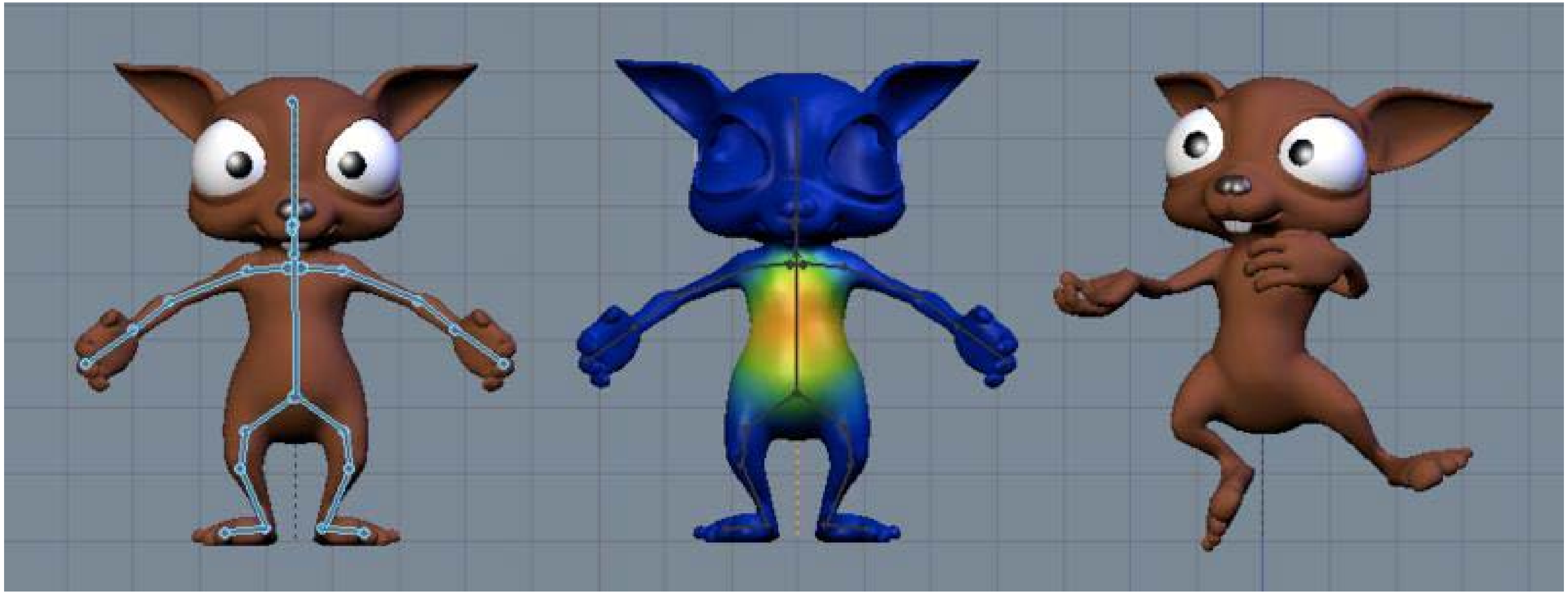
Output: Skeleton and Skinning Weights



Preliminary

Key: How to drive mesh/animate the surface

Skinning Weight: Envelop the underlying skeleton with a surface representation that conveys the appearance of the character and deforms with the underlying skeleton

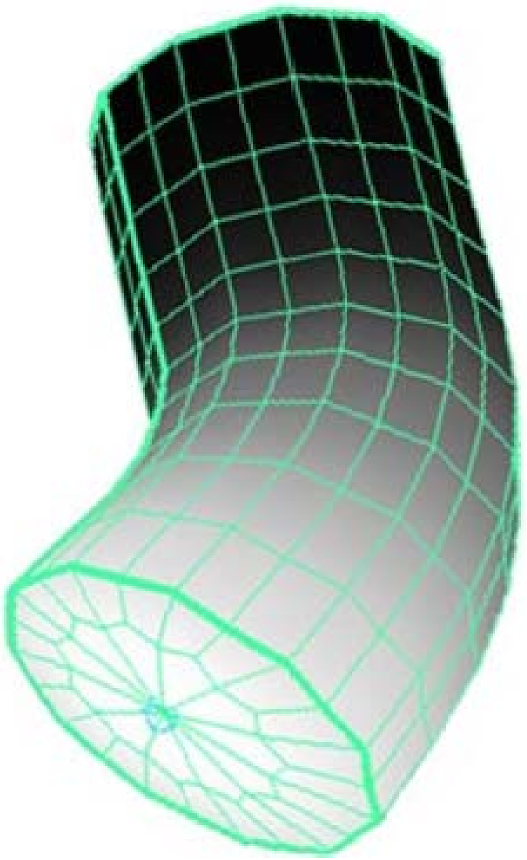


Preliminary (cont.)

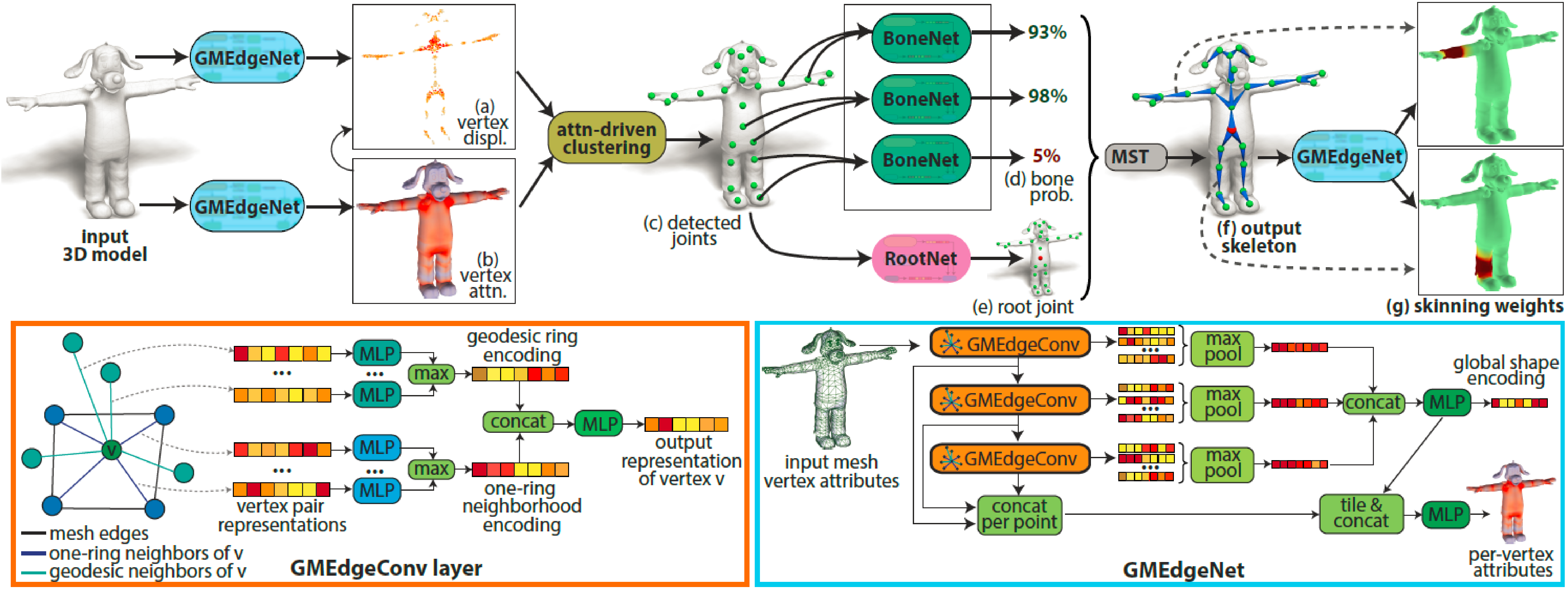
Linear Blend Skinning (LBS)

- Assign each skin vertex to more than one bone
- Each bone i to which vertex v_j belongs to is assigned a nonzero weight w_{ij}
- The world space position of the vertex is computed as the weighted average of the world space positions obtained from each bone via rigid skinning:

$$v_j' = \sum_i w_{ij} T_i v_j^i$$



System Pipeline



Skeletal joint prediction Skeleton connectivity prediction Skinning prediction

Basic Modules

GMEdgeConv layer

$$\mathbf{x}_{v,m} = \max_{u \in \mathcal{N}_m(v)} MLP(\mathbf{x}_v, \mathbf{x}_u - \mathbf{x}_v; \mathbf{w}_m)$$

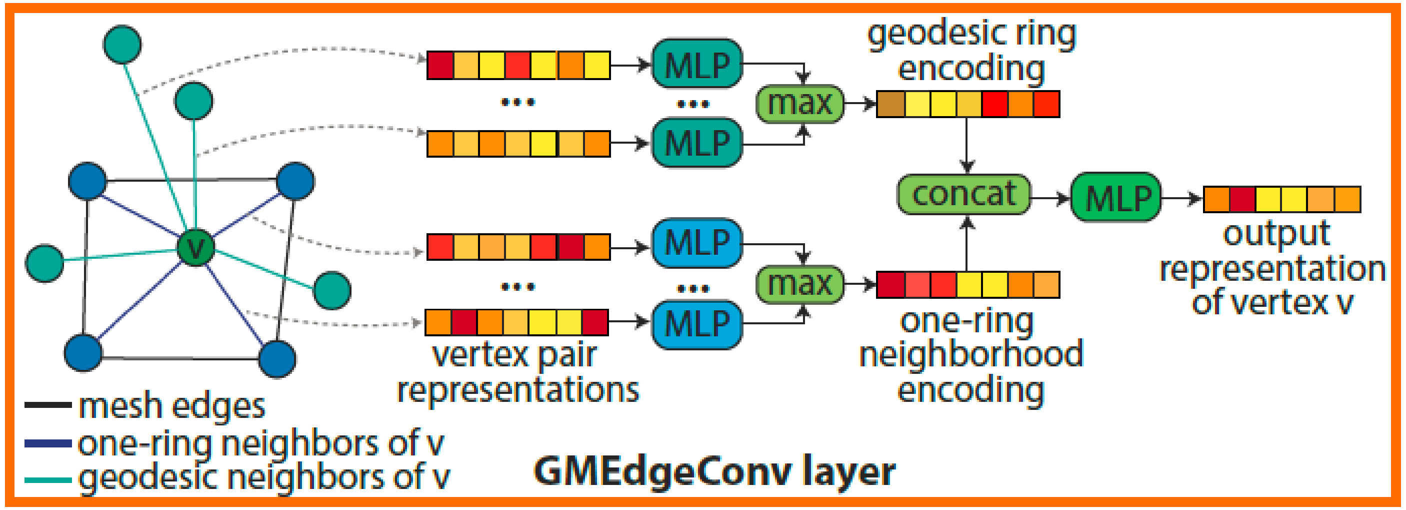
one-ring mesh neighbors

$$\mathbf{x}_{v,g} = \max_{u \in \mathcal{N}_g(v)} MLP(\mathbf{x}_v, \mathbf{x}_u - \mathbf{x}_v; \mathbf{w}_g)$$

vertices located within a geodesic ball centered at it

$$\mathbf{x}'_v = MLP(concat(\mathbf{x}_{v,m},\ \mathbf{x}_{v,g}); \mathbf{w}_c)$$

GMEdgeNet stacks three GMEdgeConv layers, each followed with a global max-pooling layer

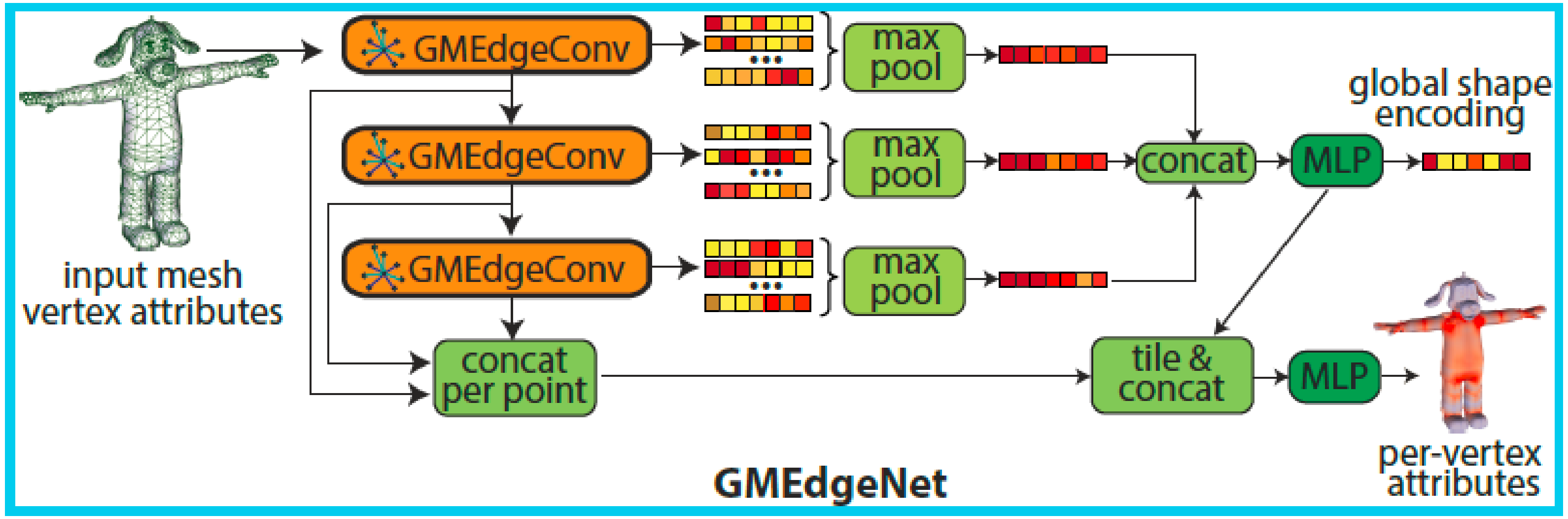


Basic Modules

GMEdgeNet

GMEdgeNet stacks three GMEdgeConv layers, each followed with a global max-pooling layer

learned vertex representations incorporate both local and global information



vertex displacement module, the feature representation are transformed to 3D displacements per each vertex through another MLP.
vertex attention module, the per-vertex feature representations are transformed through a MLP and a sigmoid nonlinearity to produce a scalar attention value per vertex

Joint Prediction

learns to **displace** mesh geometry towards candidate joint locations

Key Problem: the number of joints [not pre-defined]

combination of regression and adaptive clustering

Regression

mesh vertices are regressed to their nearest candidate joint locations

$$\mathbf{q} = \mathbf{v} + f_d(\mathcal{M}; \mathbf{w}_d)$$

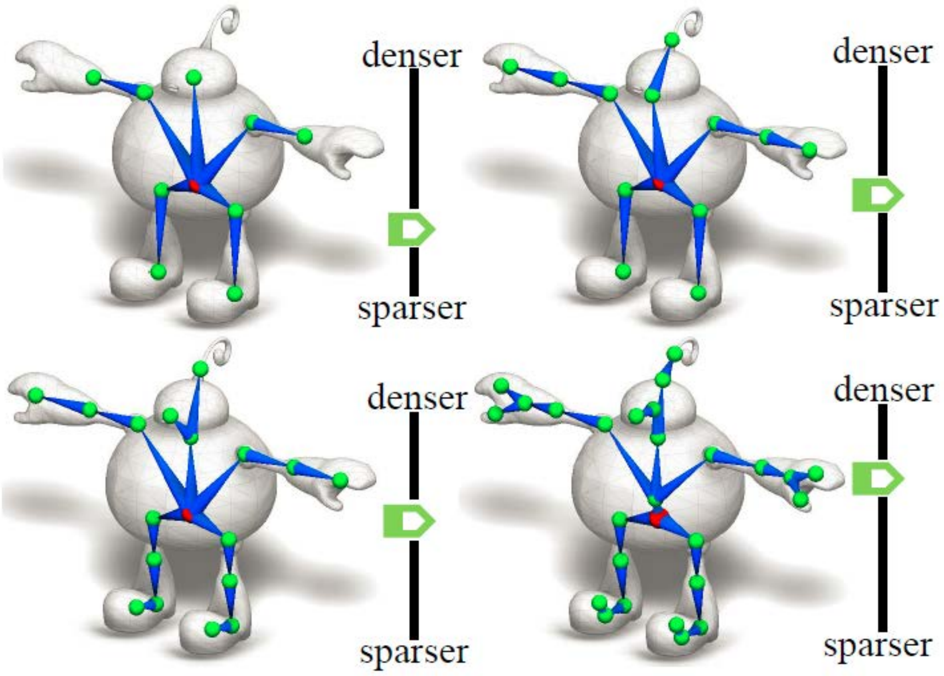
the goal is to map mesh vertices to joint locations

$$\mathbf{a} = f_a(\mathcal{M}; \mathbf{w}_a)$$

attention map includes a scalar value per vertex

Joint Prediction (cont.)

learns to **displace** mesh geometry towards candidate joint locations



Use bandwidth parameter h
that controls the level-of-detail

zero bandwidth = each displaced vertex to become a joint

Clustering

Input: displaced points $\mathbf{q}$ and attention values $\mathbf{a}$
Output: joints

variant of mean-shift clustering

Kernel:

$$\mathbf{m}_v = \frac{\sum_u a_u \cdot K(\mathbf{q}_u - \mathbf{q}_v, h) \cdot \mathbf{q}_u}{\sum_u a_u \cdot K(\mathbf{q}_u - \mathbf{q}_v, h)} - \mathbf{q}_v$$

$$K(\mathbf{q}_u - \mathbf{q}_v, h) = max(1 - ||\mathbf{q}_u - \mathbf{q}_v||^2 / h^2, 0)$$

From the largest density to create joints one by one

Symmetry as a constraint

Connectivity prediction

Bone module

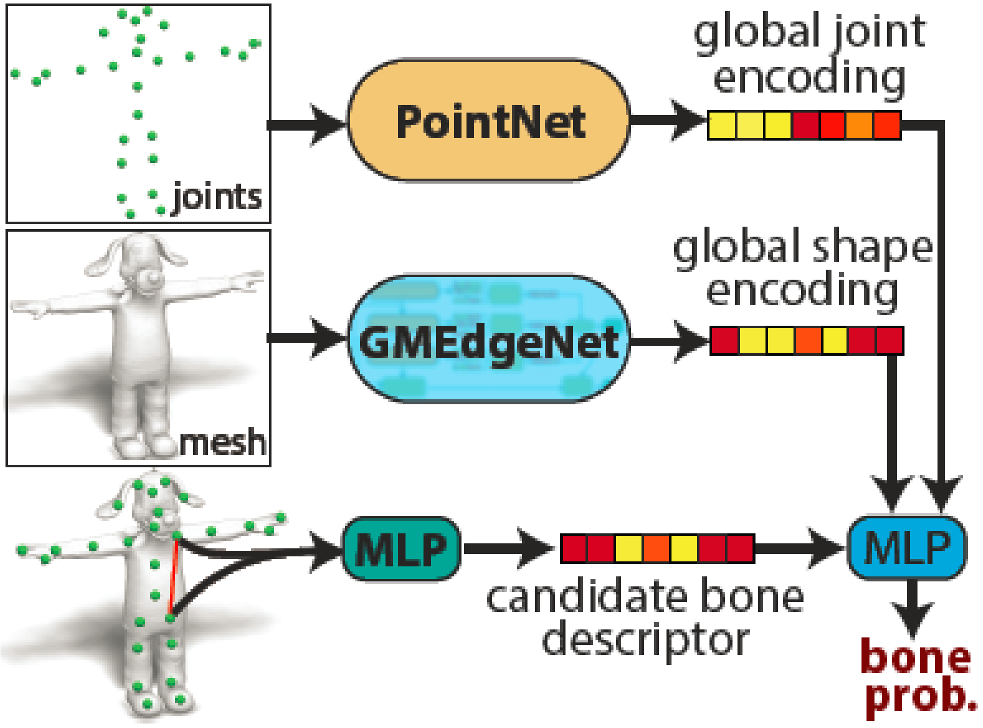
Learned neural module that outputs the **probability** of connecting each pair of joints via a bone

Inputs

(a) a 128-dimensional representation encoding the overall skeleton geometry (PointNet)

(b) a 128-dimensional representation encoding global shape geometry

(c) a representation encoding the input pair of joints $[\mathbf{t}_i, \mathbf{t}_j, d_{i,j}, o_{i,j}]$



BoneNet

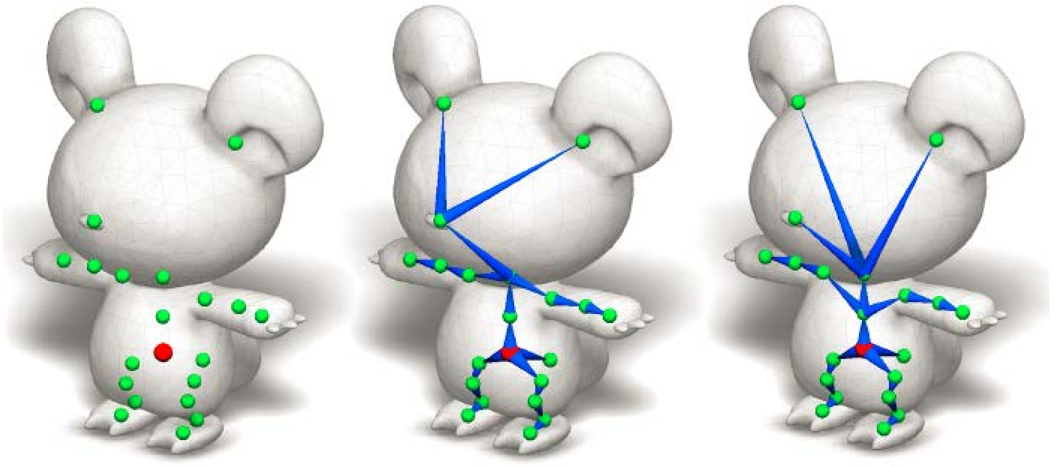
$$p_{i,j} = sigmoid\big(MLP(\mathbf{f}_{i,j}, \mathbf{g}_s, \mathbf{g}_t; \mathbf{w}_b)\big)$$

Connectivity prediction (cont.)

Skeleton extraction

$$w_{i,j} = -\log p_{i,j}$$

negative log probabilities
Weights



dense graph: nodes are the extracted joints, and edges have weights w_{ij}
use a MST algorithm to solve

Root Net

Distance to bilateral symmetry plane

$$p_{i,r} = softmax\left(MLP(\mathbf{f}_i, \mathbf{g}_s, \mathbf{g}_t; \mathbf{w}_r)\right)$$

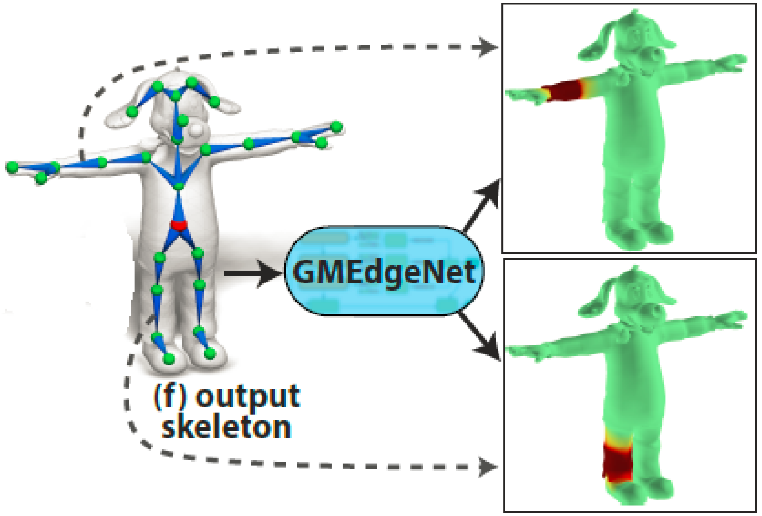
Skinning prediction

Skeleton-aware mesh representation

$\mathbf{H} = \{\mathbf{h}_v\}$

each vertex v, sort the bones according to their **volumetric geodesic distance** $\{\mathbf{b}_{r,v}\}_{r=1...K}$

5 closest bones others 0



Skinning Module

outputs a 1280-dimensional per-vertex feature vector, which is transformed to a per-vertex skinning weight vector S_v through a learned MLP and a softmax function.

Training

Joint prediction stage training

$$L_{cd}(\mathbf{w}_a, \mathbf{w}_d, h) = \frac{1}{V}\sum_{v=1}^{V} \min_k ||\mathbf{t}_v - \hat{\mathbf{t}}_k|| + \frac{1}{K}\sum_{k=1}^{K} \min_v ||\mathbf{t}_v - \hat{\mathbf{t}}_k||$$

Chamfer distance between collapsed vertices $\{\mathbf{t}_v\}$ and training joints $\{\hat{\mathbf{t}}_k\}$

$$L'_{cd}(\mathbf{w}_d) = \frac{1}{V}\sum_{v} \min_k ||\mathbf{q}_v - \hat{\mathbf{t}}_k|| + \frac{1}{K}\sum_{k} \min_v ||\mathbf{q}_v - \hat{\mathbf{t}}_k||$$

Supervised vertex displacements

$$L_m(\mathbf{w}_a) = \hat{\mathbf{m}} \log \mathbf{a} + (1 - \hat{\mathbf{m}}) \log(1 - \mathbf{a})$$

cross-entropy between these masks and neural attention

binary mask for attention map

Connectivity stage training

$$L_m(\mathbf{w}_a) = \sum_{i,j} \hat{p}_{ij} \log p_{i,j} + (1 - \hat{p}_{ij}) \log(1 - p_{i,j})$$

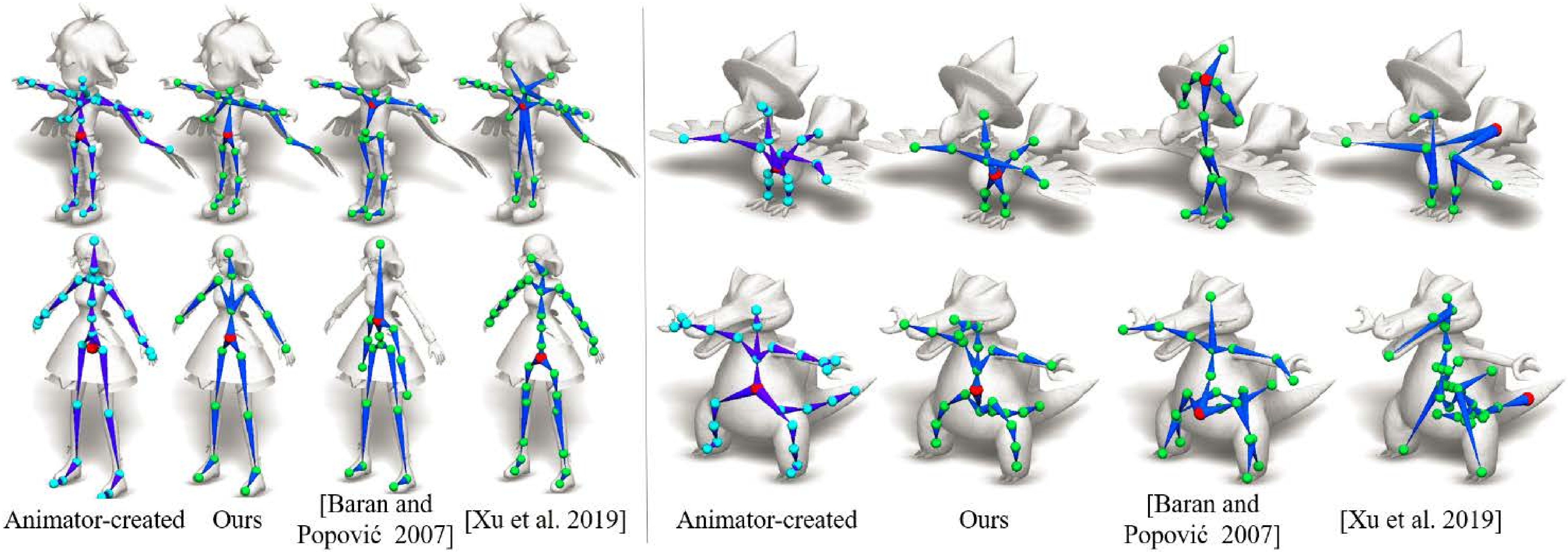
cross-entropy between the training adjacency matrix entries and the predicted probabilities

Skinning stage training

$$L_s(\mathbf{w}_s) = \frac{1}{V}\sum_{v}\sum_{r} \hat{s}_{v,r} \log s_{v,r}$$

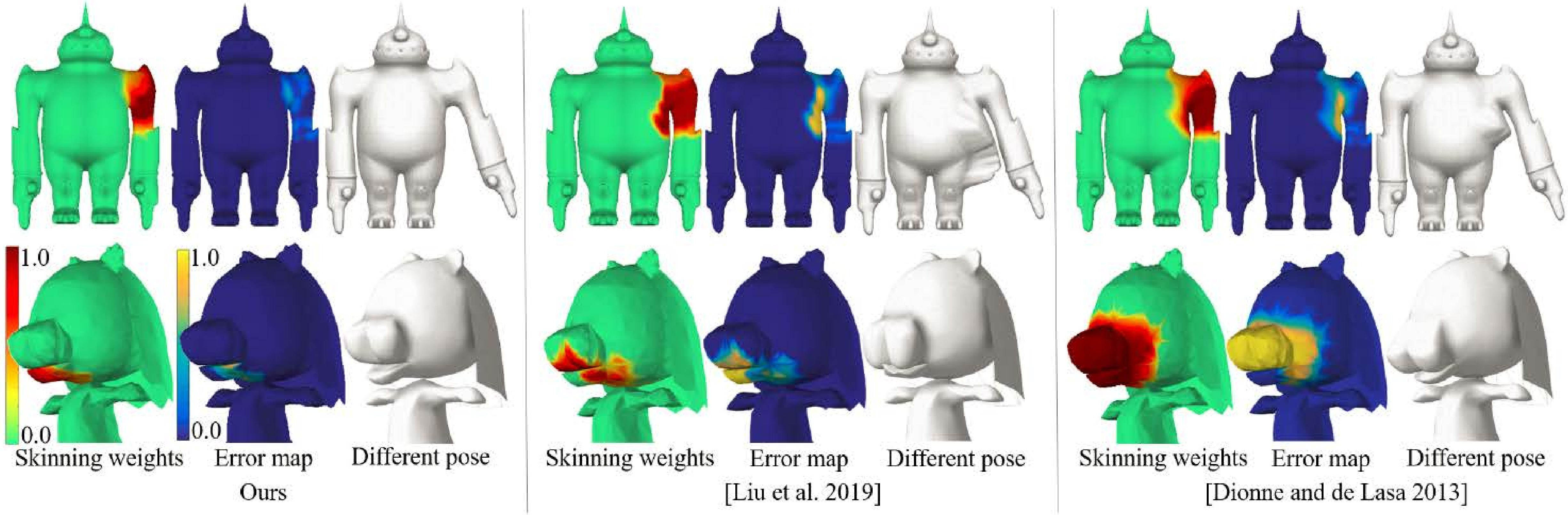
cross-entropy training and predicted distributions for each vertex

Results



skeleton extraction results comparison

Results



Skinning results comparison

Results

	IoU	Prec.	Rec.	CD-J2J	CD-J2B	CD-B2B
Pinocchio	36.5%	38.7%	35.9%	7.2%	5.5%	4.7%
Xu et al. 2019	53.7%	53.9%	55.2%	4.5%	2.9%	2.6%
Ours	**61.6%**	**67.6%**	**58.9%**	**3.9%**	**2.4%**	**2.2%**

Table 1. Comparisons with other skeleton prediction methods.

	Prec.	Rec.	avg L1	avg dist	max dist
BBW	68.3%	77.6 %	0.69	0.0061	0.055
GeoVoxel	72.8%	75.1 %	0.65	0.0057	0.049
NeuroSkinning	76.3%	74.7 %	0.57	0.0053	0.043
Ours	**82.3%**	**80.8%**	**0.39**	**0.0041**	**0.032**

Table 2. Comparisons with other skinning prediction methods.

Ablation

	IoU	Prec.	Rec.	CD-J2J	CD-J2B	CD-B2B
P2PNet-based	40.6%	41.6%	42.0%	6.3%	4.6%	3.8%
No attn	52.4%	50.9%	50.7%	4.6%	3.1%	2.7%
One-ring	59.7%	65.6%	57.4%	4.1%	2.5%	2.4%
No vertex loss	59.3%	58.2%	57.6%	4.2%	2.7%	2.5%
No attn pretrain	60.6%	64.0%	58.1%	4.2%	2.6%	2.4%
Full	**61.6%**	**67.6%**	**58.9%**	**3.9%**	**2.4%**	**2.2%**

Table 3. Joint prediction ablation study

	Class. Acc.	CD-B2B	ED
Euclidean edge cost	61.2%	0.30%	5.0
bone descriptor only	71.9%	0.22%	4.2
bone descriptor+skel. geometry	80.7%	0.12%	2.9
Full stage	**83.7%**	**0.10%**	**2.4**

Table 4. Connectivity prediction ablation study

	Prec	Rec.	avg-L1	avg-dist.	max-dist.
No geod. dist.	80.0%	79.3%	0.41	0.0044	0.054
Ours	**82.3%**	**80.8%**	**0.39**	**0.0041**	**0.032**

Table 5. Skinning prediction ablation study

Weakness:
1. input training and test shapes have a **consistent upright** , **front facing orientation, and T-pose**

2. mesh resolution should near the training dataset

3. connectivity is not guaranteed