Learning 3D Shape Fidelity Metric from Real-world Distortions

Supplementary Material

1. Evaluation methods

We employ three distinct evaluation methods to assess the correlation between our proposed metrics and human scoring (ground truth) on the Real Shape Fidelity Dataset we provide.

Pearson's Linear Correlation Coefficient (PLCC). PLCC measures the linear alignment between our proposed metrics and human evaluation. The definition is denoted as

$$p = \frac{\sum_{i=1}^{n}(\hat{s}_i - \bar{\hat{s}})(s_i - \bar{s})}{\sqrt{\sum_{i=1}^{n}(\hat{s}_i - \bar{\hat{s}})^2}\sqrt{\sum_{i=1}^{n}(s_i - \bar{s})^2}}, \tag{1}$$

where $\hat{s}_i$ and s_i denote the predicted and ground-truth fidelity scores of the sample i, respectively, n is the total number of samples. Additionally, $\bar{\hat{s}} = \frac{1}{n}\sum_{i=1}^{n}\hat{s}_i$ and $\bar{s} = \frac{1}{n}\sum_{i=1}^{n}s_i$ are the average predicted score and ground-truth score, respectively.

Spearman's rank order correlation coefficient (SROCC). SROCC measures the strength and direction of the monotonic relationship between two variables by computing the Pearson correlation between their ranked values. It considers the magnitude of rank differences, making it sensitive to non-linear correlations.

$$r_s = 1 - \frac{6\sum_{i=1}^{n}(R(\hat{s}_i) - R(s_i))^2}{n(n^2-1)}, \tag{2}$$

where $R(\hat{s}_i)$ and $R(s_i)$ represent the rankings of $\hat{s}_i$ and s_i, respectively, and n denotes the total number of data points. In our paper, n corresponds to the number of meshes scored by a single subject.

Kendall's rank order correlation coefficient (KROCC). KROCC quantifies the association between two variables by counting the number of concordant and discordant pairs in their rankings. Unlike SROCC, it only evaluates the relative order of ranks, making it more robust to outliers and slight variations.

$$\tau = 1 - \frac{2}{n(n^2-1)}\sum_{i<j}\operatorname{sgn}(\hat{s}_i - \hat{s}_j))\operatorname{sgn}(s_i - s_j). \tag{3}$$

The function $\operatorname{sgn}(\cdot)$ represents the sign function, defined as follows:

$$\operatorname{sgn}(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ 0, & x = 0 \end{cases}$$

The key distinction between Spearman's Rank Order Correlation Coefficient (SROCC) and Kendall's Rank Order Correlation Coefficient (KROCC) lies in their approach to measuring rank correlation. SROCC accounts for the actual magnitude of rank differences in the input data, whereas KROCC only considers the number of discordant (inverted) pairs.

2. Proof of translation, rotation, and scale invariance

We provide theoretical justification for the effectiveness of our Invariance Alignment Module by proving that, under any global translation, rotation, or isotropic scaling transformation applied to a 3D mesh, the output of our alignment procedure remains unchanged.

Let the original input mesh have N vertices denoted by $\{\mathbf{v}_i\}_{i=1}^{N}$, where $\mathbf{v}_i \in \mathbb{R}^3$. Consider a transformed version of this mesh defined as:

$$\mathbf{v}_i^{\text{trans}} = sR\mathbf{v}_i + \mathbf{t}, \quad \forall i, \tag{4}$$

where $s > 0$ is a scalar scale factor, $R \in \mathbb{R}^{3\times3}$ is an orthonormal rotation matrix ($R^\top R = I$), and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector.

We show that applying our alignment module to both the original mesh $\{\mathbf{v}_i\}$ and the transformed mesh $\{\mathbf{v}_i^{\text{trans}}\}$ yields the same normalized result.

Translation invariance. Let $\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i$ and $\mathbf{c}^{\text{trans}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i^{\text{trans}}$. By substituting Eq. (4):

$$\mathbf{c}^{\text{trans}} = \frac{1}{N}\sum_{i=1}^{N}(sR\mathbf{v}_i + \mathbf{t}) = sR\mathbf{c} + \mathbf{t}. \tag{5}$$

Thus, subtracting the centroid gives:

$$\mathbf{v}_i^{\text{trans}} - \mathbf{c}^{\text{trans}} = sR\mathbf{v}_i + \mathbf{t} - (sR\mathbf{c} + \mathbf{t}) \tag{6}$$

$$= sR(\mathbf{v}_i - \mathbf{c}). \tag{7}$$

This proves that translation is eliminated, and the centered mesh depends only on the original vertex offsets.

Rotation invariance. Define $V = [\mathbf{v}_1 - \mathbf{c}, \ldots, \mathbf{v}_N - \mathbf{c}]^\top \in \mathbb{R}^{N\times3}$ as the centered vertex matrix. For the transformed mesh, we have:

$$V^{\text{trans}} = sVR^\top. \tag{8}$$

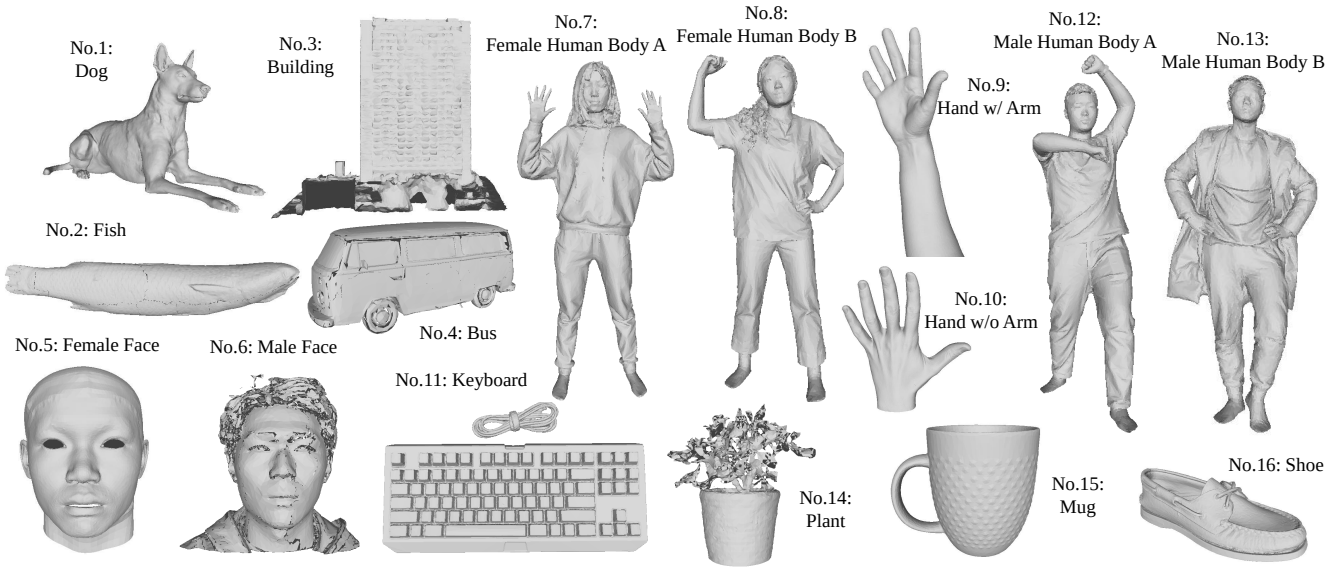

Figure 1. Objects in our provided Real Shape Fidelity main subset and what the object numbers correspond to in the main paper Tab. 1, 2, and 3.

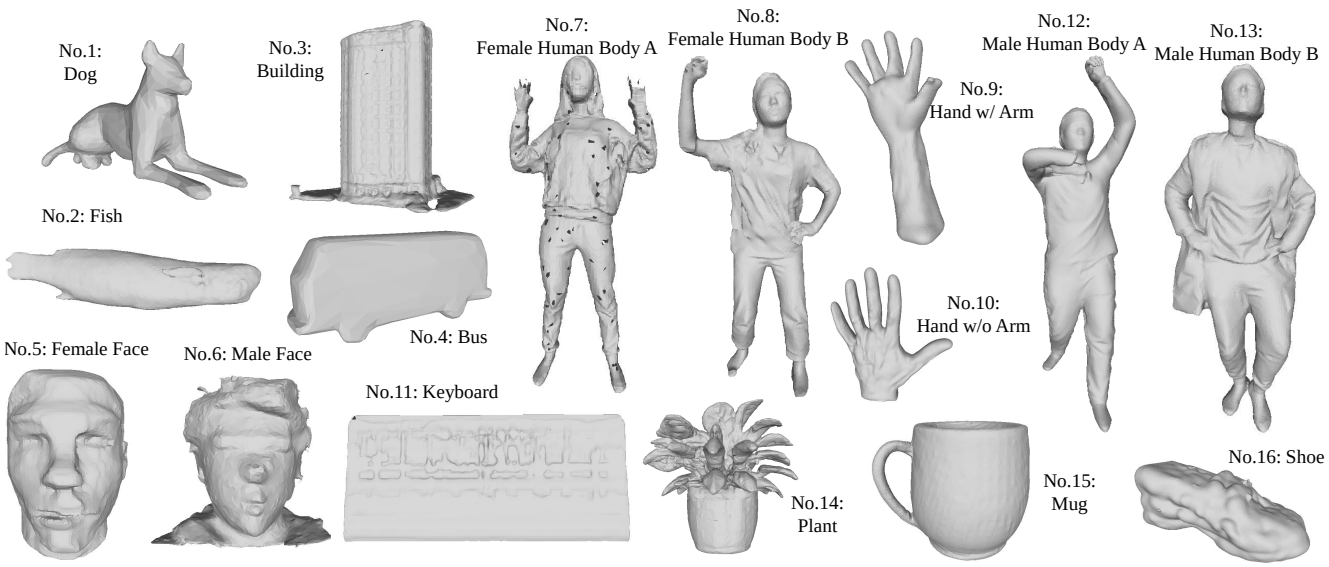

Figure 2. We visualize examples of distorted meshes of different generation methods in our provided Real Shape Fidelity main subset.

The covariance matrix becomes:

$$\Sigma^{\text{trans}} = \frac{1}{N}(V^{\text{trans}})^\top V^{\text{trans}} = \frac{1}{N}(sRV^\top)(sVR^\top) \quad (9)$$

$$= s^2 R\left(\frac{1}{N}V^\top V\right)R^\top = R\Sigma R^\top. \quad (10)$$

This shows that the eigenvalues of Σ are preserved, and the eigenvectors of Σ^{trans} are the rotated versions of Σ. Therefore, applying the eigenvector alignment step (PCA) will bring both versions to the same canonical orientation:

$$U_{\text{trans}}^\top(sR(\mathbf{v}_i - \mathbf{c})) = U^\top(\mathbf{v}_i - \mathbf{c}), \quad (11)$$

where $U_{\text{trans}} = RU$ and U is the PCA basis of the original mesh.

Scale invariance. After centering and rotating, the transformed vertex becomes:

$$\mathbf{v}_i''^{\text{trans}} = s \cdot U^\top(\mathbf{v}_i - \mathbf{c}). \quad (12)$$

The average norm (used for scale normalization) is:

$$r^{\text{trans}} = \frac{1}{N}\sum_{i=1}^{N} \|s \cdot U^\top(\mathbf{v}_i - \mathbf{c})\|_2 = s \cdot r, \quad (13)$$

where r is the original average distance to the origin. After scale normalization:

$$\frac{1}{r^{\text{trans}}} \cdot \mathbf{v}_i''^{\text{trans}} = \frac{1}{sr} \cdot s \cdot U^\top (\mathbf{v}_i - \mathbf{c}) = \frac{1}{r} \cdot U^\top (\mathbf{v}_i - \mathbf{c}). \quad (14)$$

Thus, the final output is identical for both original and transformed meshes.

Conclusion. The output of our alignment module is invariant to any global similarity transformation (translation, rotation, and scale). Hence, our metric can evaluate shape fidelity independently of these geometric variations.

3. Detailed loss designs

Smooth L1 loss. The smooth L1 loss ensures the predicted score is close to the labeled score. It is defined as:

$$\mathcal{L}_{\text{smooth}} = \sum_{i=1}^{n} \begin{cases} \frac{1}{2}(\hat{s}_i - s_i)^2, & \text{if } |\hat{s}_i - s_i| < 1 \\ |\hat{s}_i - s_i| - \frac{1}{2}, & \text{otherwise} \end{cases} \quad (15)$$

where $\hat{s}_i$ and s_i denote the predicted and ground truth fidelity scores of the sample i, respectively, n is the total number of samples.

Pearson's correlation loss. Pearson's correlation loss encourages a stronger linear correlation between the predicted and ground truth scores. It is defined as:

$$\mathcal{L}_{\text{plcc}} = \frac{\sum_{i=1}^{n}(\hat{s}_i - \bar{\hat{s}})(s_i - \bar{s})}{\sqrt{\sum_{i=1}^{n}(\hat{s}_i - \bar{\hat{s}})^2}\sqrt{\sum_{i=1}^{n}(s_i - \bar{s})^2}}, \quad (16)$$

where $\bar{\hat{s}} = \frac{1}{n}\sum_{i=1}^{n}\hat{s}_i$ and $\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s_i$ are the average predicted score and ground truth score, respectively.

Spearman's ranking order loss. The Spearman's ranking order loss encourages a higher ranking correlation between the predicted and ground truth scores. It is defined as:

$$\mathcal{L}_{\text{srocc}} = 1 - \frac{6\sum_{i=1}^{n}(R(\hat{s}_i) - R(s_i))^2}{n(n^2-1)}, \quad (17)$$

where $R(\cdot)$ denotes the ranking order. Because Eq. (17) is not inherently differentiable, we use the differentiable ranking approach proposed in [2] to make it differentiable for optimization.

4. Real Shape Fidelity dataset

We present all objects in our Real Shape Fidelity main subset in Fig. 1 and a corresponding distorted mesh for each object in Fig. 2. Additionally, Tab. 1 and Tab. 2 provide an overview of the methods used to generate these distorted meshes. Specifically, the objects in our dataset are collected from the following sources: RenderBot [11] (Dog), ARC3D [1] (Fish, Bus, Hand), Google Scanned Objects [3] (Keyboard, Mug, Shoe), Sketchfab [14] (Plant), FaceScape [28, 31] (Female Face), FaceVerse [21] (Male Face), and Function4D [29] (Female and Male Human Bodies). The distorted meshes in Real Shape Fidelity main subset are generated using a wide range of methods, including: CRM [22], DreamGaussian [16], InstantMesh [26], LGM [17], One2345 [8], One2345pp[9], PeRFlowText[27], ShapE [6], ShapE-Text [6], SplatterImage [15], TripoSR [19], ECON [25], ICON [24], PiFU [12], PiFUHD [13], and HaMeR [10].

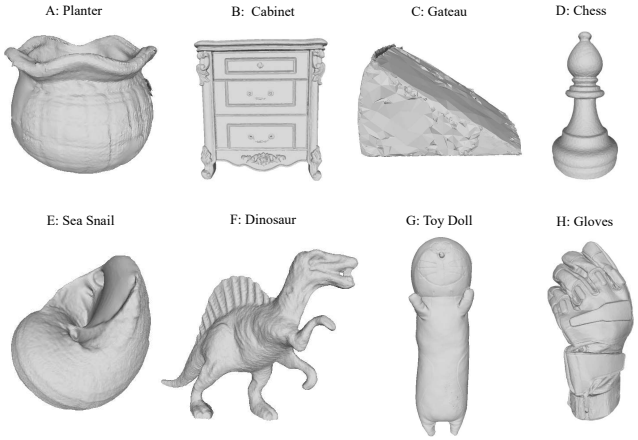

Figure 3. Objects in our provided Real Shape Fidelity test-only subset and what the object numbers correspond to in the main paper Tab. 5, 6, and 7.

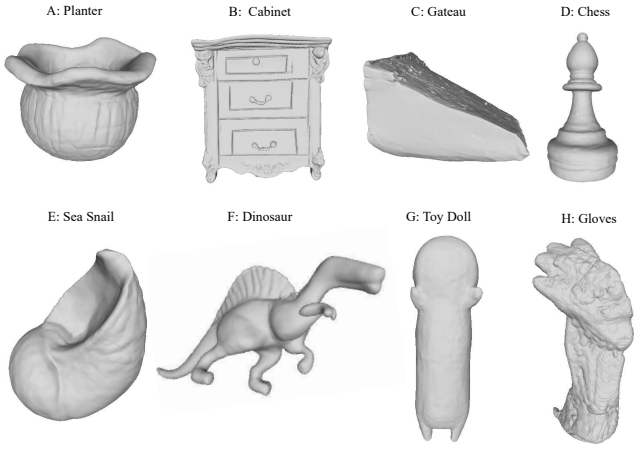

Figure 4. We visualize examples of distorted meshes of different generation methods in our provided Real Shape Fidelity test-only subset.

For Real Shape Fidelity test-only subset, Fig. 3 and Fig. 4 show the object names, referenced objects, some of the distortions. Specifically, the unseen distortions are produced by five recent 3D generation methods—CraftsMan3D [7], Hunyuan3D 2.1 [18], SPAR3D [5], TripoSR [20], and InstantMesh [26]—most of which are not included in training. The evaluation is further conducted on eight unseen objects from OmniObject3D [23], none of

Table 1. Distribution of distorted mesh generation methods in our dataset. A checkmark (✓) indicates that the corresponding method generated the distorted mesh for a specific object category. (Part 1)

Object Type	CRM	DreamGaussian	InstantMesh	LGM	One2345	One2345pp	PeRFlowText	ShapE
Dog [11]	✓	✓	✓	✓	✓	✓	✓	✓
Fish [1]	✓	✓	✓	✓	✓	✓	✓	✓
Building	✓	✓	✓			✓	✓	✓
Bus [1]	✓	✓	✓	✓		✓	✓	✓
Female Face [28, 31]	✓	✓	✓	✓	✓		✓	
Male Face [21]	✓	✓	✓	✓	✓		✓	
Female Human Body A [29]	✓	✓	✓	✓		✓	✓	✓
Female Human Body B [29]	✓	✓	✓	✓		✓	✓	✓
Hand w/ Arm [1]	✓	✓	✓	✓	✓		✓	✓
Hand w/o Arm [1]	✓	✓	✓	✓	✓		✓	✓
Keyboard [3]	✓	✓	✓	✓		✓	✓	✓
Male Human Body A [29]	✓	✓	✓	✓		✓	✓	✓
Male Human Body B [29]	✓	✓	✓	✓		✓	✓	✓
Mug [3]	✓	✓	✓	✓	✓	✓	✓	✓
Plant [14]	✓	✓	✓	✓		✓	✓	✓
Shoe [3]	✓	✓	✓	✓	✓	✓	✓	✓

Table 2. Distribution of distorted mesh generation methods in our dataset. A checkmark (✓) indicates that the corresponding method generated the distorted mesh for a specific object category. (Part 2)

Object Type	ShapEText	SplatterImage	TripoSR	ECON	ICON	PiFU	PiFUHD	HaMeR
Dog [11]	✓	✓	✓					
Fish [1]	✓	✓	✓					
Building	✓	✓	✓					
Bus [1]	✓		✓					
Female Face [28, 31]	✓	✓	✓					
Male Face [21]	✓	✓	✓					
Female Human Body A [29]	✓	✓	✓	✓	✓	✓	✓	
Female Human Body B [29]	✓	✓	✓	✓	✓	✓	✓	
Hand w/ Arm [1]	✓		✓					✓
Hand w/o Arm [1]	✓		✓					✓
Keyboard [3]	✓	✓	✓					
Male Human Body A [29]	✓	✓	✓	✓	✓	✓	✓	
Male Human Body B [29]	✓	✓	✓	✓	✓	✓	✓	
Mug [3]	✓		✓					
Plant [14]	✓	✓	✓					
Shoe [3]	✓	✓	✓					

which appear in the training set. About 500 online crowd workers participated in the annotation; scores were obtained via a Swiss tournament.

5. Example of neighbor vertices

We show how the selected 64 neighbors are distributed around the central vertices. The red vertex is the central vertex, and the blue vertices are the selected 64 nearest neighbors. As shown in Fig. 5, we can observe that the spatial range of the neighbor vertices includes the details.

6. Ablations and Analyses

6.1. Metric stableness using various training set

To assess the stability of proposed metric, we design two experiments in which we calculate the stableness of our metric trained on various training set. In the first experiment, we selected 5 objects. For each of these objects, we computed pairwise cosine similarity between the 11 models trained on folds that include the same 5 objects as training data, resulting in 11×11 cosine similarity matrices. Then, we calculated the element-wise average and standard deviation of the cosine similarity matrices across the 5 objects (result in

Table 3. Mean cosine similarity matrix on object No.1-5

	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12	Model 13	Model 14	Model 15	Model 16
Model 6	1.0000	0.9555	0.9492	0.9381	0.9544	0.8618	0.9519	0.9413	0.9220	0.9326	0.9589
Model 7	0.9555	1.0000	0.9983	0.9941	0.9996	0.9288	0.9966	0.9967	0.9780	0.9906	0.9904
Model 8	0.9492	0.9983	1.0000	0.9975	0.9987	0.9326	0.9976	0.9986	0.9732	0.9941	0.9866
Model 9	0.9381	0.9941	0.9975	1.0000	0.9953	0.9364	0.9968	0.9967	0.9641	0.9950	0.9811
Model 10	0.9544	0.9996	0.9987	0.9953	1.0000	0.9307	0.9976	0.9974	0.9764	0.9926	0.9901
Model 11	0.8618	0.9288	0.9326	0.9364	0.9307	1.0000	0.9330	0.9303	0.9108	0.9416	0.9030
Model 12	0.9519	0.9966	0.9976	0.9968	0.9976	0.9330	1.0000	0.9955	0.9684	0.9941	0.9885
Model 13	0.9413	0.9967	0.9986	0.9967	0.9974	0.9303	0.9955	1.0000	0.9743	0.9929	0.9826
Model 14	0.9220	0.9780	0.9732	0.9641	0.9764	0.9108	0.9684	0.9743	1.0000	0.9642	0.9687
Model 15	0.9326	0.9906	0.9941	0.9950	0.9926	0.9416	0.9941	0.9929	0.9642	1.0000	0.9789
Model 16	0.9589	0.9904	0.9866	0.9811	0.9901	0.9030	0.9885	0.9826	0.9687	0.9789	1.0000

Table 4. Standard derivation of cosine similarity matrix among object No.1-5

	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12	Model 13	Model 14	Model 15	Model 16
Model 6	0.0000	0.0192	0.0223	0.0327	0.0187	0.0796	0.0226	0.0306	0.0330	0.0264	0.0078
Model 7	0.0192	0.0000	0.0009	0.0061	0.0002	0.0676	0.0011	0.0014	0.0158	0.0041	0.0084
Model 8	0.0223	0.0009	0.0000	0.0017	0.0003	0.0707	0.0013	0.0008	0.0207	0.0046	0.0097
Model 9	0.0327	0.0061	0.0017	0.0000	0.0043	0.0658	0.0016	0.0013	0.0312	0.0035	0.0118
Model 10	0.0187	0.0002	0.0003	0.0043	0.0000	0.0654	0.0005	0.0017	0.0179	0.0033	0.0080
Model 11	0.0796	0.0676	0.0707	0.0658	0.0654	0.0000	0.0598	0.0747	0.0561	0.0571	0.0658
Model 12	0.0226	0.0011	0.0013	0.0016	0.0005	0.0598	0.0000	0.0028	0.0245	0.0031	0.0068
Model 13	0.0306	0.0014	0.0008	0.0013	0.0017	0.0747	0.0028	0.0000	0.0200	0.0071	0.0133
Model 14	0.0330	0.0158	0.0207	0.0312	0.0179	0.0561	0.0245	0.0200	0.0000	0.0298	0.0108
Model 15	0.0264	0.0041	0.0046	0.0035	0.0033	0.0571	0.0031	0.0071	0.0298	0.0000	0.0120
Model 16	0.0078	0.0084	0.0097	0.0118	0.0080	0.0658	0.0068	0.0133	0.0108	0.0120	0.0000

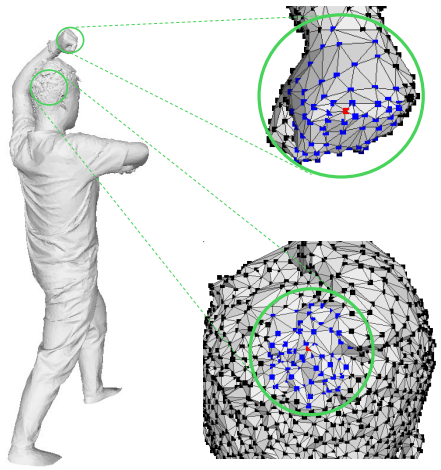

Figure 5. Examples of the selected neighbor vertices. The red vertex in each example is the central vertex, and the blue vertices are the selected 64 nearest-neighbor vertices.

Tab. 3 and Tab. 4.).

In the second experiment, we repeated a similar process and computed pairwise cosine similarity between the 5 models trained on folds that include the same 11 objects as training data, resulting in 5x5 cosine similarity matrices. Again, we computed element-wise averages and standard deviations across the 11 objects (result in Tab. 5 and Tab. 6.).

This setup allows us to assess the consistency of the metric across different folds for the same object (mean), and the stability of this consistency across different objects (standard deviation). In the ideal case, if all models across folds behave identically for a given object, each entry of cosine similarity matrix would be 1, and the standard deviation matrix would be all 0.

For object selection, we simply used the first 5 objects (No. 1–5) and the first 11 objects (No. 1–11) for the two experiments, respectively. From the results, we observe that in both experiments, most entries in the cosine similarity matrices have means above 0.95, and most entries in the standard deviation matrices are below 0.05. These findings demonstrate the consistency and stability of our metric across different folds during k-fold training.

Table 5. Mean cosine similarity matrix on object No.1-11

	Model 12	Model 13	Model 14	Model 15	Model 16
Model 12	1.0000	0.9851	0.9675	0.9783	0.9779
Model 13	0.9851	1.0000	0.9835	0.9803	0.9855
Model 14	0.9675	0.9835	1.0000	0.9626	0.9754
Model 15	0.9783	0.9803	0.9626	1.0000	0.9813
Model 16	0.9779	0.9855	0.9754	0.9813	1.0000

Table 6. Standard derivation of cosine similarity matrix among object No.1-11

	Model 12	Model 13	Model 14	Model 15	Model 16
Model 12	0.0000	0.0286	0.0534	0.0352	0.0299
Model 13	0.0286	0.0000	0.0268	0.0419	0.0276
Model 14	0.0534	0.0268	0.0000	0.0766	0.0430
Model 15	0.0352	0.0419	0.0766	0.0000	0.0357
Model 16	0.0299	0.0276	0.0430	0.0357	0.0000

6.2. Data sufficiency and generalizability

Although the dataset contains only 16 objects, the supervision is defined on distorted–GT mesh pairs, encouraging the model to learn distortion-related geometric patterns rather than object-specific appearance cues. We further reduce the risk of overfitting by adopting a lightweight PointNet-style architecture. In the main paper, we already validate the generalization ability of our model through K-fold cross-validation (Tabs. 1–3) and out-of-domain test-only evaluation (Tab. 5). In addition, Tabs. 3–6 in the supplementary material show that the influence of individual training samples on performance stability is limited.

To further examine whether the current data volume is sufficient, we train the model with 50%, 100%, and 200% of the training data, where 200% is obtained via data augmentation. As shown in Tab. 7, reducing the training data to 50% leads to a clear performance drop, while expanding the data to 200% brings only marginal improvement over the original setting. This suggests that the current data volume is already sufficient for the proposed model. At the same time, we agree that with more real data available in the future, larger-capacity models may further improve performance.

Table 7. Effect of training data volume on LoCaSE. Using only 50% of the training data causes a clear performance drop, while increasing the data volume to 200% via augmentation yields only marginal gains, suggesting that the current data scale is sufficient for the proposed model.

Method	PLCC ↑	SROCC ↑	KROCC ↑
50% data	0.358	0.500	0.358
100% data	0.728	**0.757**	**0.614**
200% data	**0.735**	0.728	0.530

Table 8. Comparison with image-based baselines. LoCaSE achieves substantially higher correlation with human judgments than LPIPS and DreamSim.

Method	PLCC ↑	SROCC ↑	KROCC ↑
LPIPS (12 views)	0.518	0.483	0.331
DreamSim (12 views)	0.705	0.681	0.490
LoCaSE	**0.728**	**0.757**	**0.614**

6.3. Discretization Invariance

We evaluate the discretization invariance of LoCaSE by applying both extreme and realistic modifications to the input meshes, including triangle soup conversion, mesh subdivision, and additive vertex noise. As shown in Tab. 9, the performance remains largely stable across all settings. This indicates that LoCaSE primarily captures geometric fidelity rather than relying on mesh connectivity patterns or a specific discretization. Overall, the proposed metric is robust to variations in mesh resolution and topological structure.

Table 9. Discretization invariance analysis of LoCaSE under different mesh discretization changes. The consistently stable performance across triangle soup conversion, subdivision, and vertex perturbation demonstrates that LoCaSE mainly captures geometric fidelity rather than depending on a particular mesh connectivity or discretization pattern.

Setting	PLCC ↑	SROCC ↑	KROCC ↑	ΔPLCC
Triangle soup	0.728	0.758	0.615	-0.03%
Subdivide (1 iter)	0.728	0.754	0.610	+0.04%
Subdivide (2 iter)	0.752	0.757	0.616	+3.26%
Noise ($\sigma = 0.001$)	0.727	0.747	0.597	-0.11%
Noise ($\sigma = 0.01$)	0.725	0.745	0.593	-0.44%

6.4. Stability Under PCA Variations

To assess the effect of PCA-induced instability, we repeatedly apply several symmetry-preserving or near-symmetry transformations to the meshes, including sign flipping, axis swapping, rotation, and small perturbations, and measure the resulting variation in LoCaSE. As shown in Tab. 10, these PCA-related variations lead to only minor changes in the metric output. This suggests that LoCaSE is stable under common PCA ambiguities and perturbations.

Table 10. Stability analysis of LoCaSE under PCA-related variations. The low standard deviation across sign flipping, axis swapping, rotation, and small perturbations indicates that PCA-induced instability has only a minor effect on the metric.

Experiment	Std ↓
Sign flipping	0.031
Axis swapping	0.052
Rotation consistency	0.040
Small perturbation	0.005

6.5. Robustness to Geometric Noise

LoCaSE is designed to be stable under realistic geometric perturbations. To evaluate this property, we measure the average change in the metric output under several types of noise, including rotation, outliers, scaling, and vertex perturbation. As shown in Tab. 11, LoCaSE remains highly stable under scale changes and small vertex noise, while showing only moderate variation under larger rotations and higher outlier ratios. Overall, these results indicate that Lo-

CaSE is robust to a wide range of realistic geometric disturbances.

Table 11. Robustness of LoCaSE under different geometric perturbations. We report the average relative change in metric output (Δ) under varying levels of rotation, outliers, scaling, and vertex noise. LoCaSE remains highly stable under most realistic perturbations.

Noise Level	Avg Δ	Noise Level	Avg Δ
Rotation / 1°	1.87%	Outliers / 0.5%	1.64%
Rotation / 2°	2.39%	Outliers / 1.0%	1.61%
Rotation / 5°	3.96%	Outliers / 2.0%	1.80%
Rotation / 10°	6.42%	Outliers / 5.0%	2.15%
Scale / 0.5%	0.02%	Noise / 0.05%	0.08%
Scale / 1.0%	0.02%	Noise / 0.10%	0.17%
Scale / 2.0%	0.01%	Noise / 0.50%	1.04%
Scale / 5.0%	0.02%	Noise / 1.00%	2.42%

6.6. Comparison with image-based baselines

We also compare our method with representative image-based perceptual metrics, including LPIPS [30] and DreamSim [4]. For fairness, all methods are evaluated under the same protocol. As shown in Tab. 8, LoCaSE consistently achieves higher correlation with human ratings than both image-based baselines, demonstrating the advantage of geometry-aware quality assessment for distorted 3D meshes.

6.7. Evaluation on the Shape Grading Dataset

We further evaluate LoCaSE on the synthetic Shape Grading dataset to examine its generalizability beyond our main benchmark. As shown in Tab. 12, compared with SAUCD, LoCaSE obtains slightly lower PLCC but higher rank correlations in both SROCC and KROCC. These results indicate that LoCaSE generalizes well to synthetic data and remains competitive on a dataset with different distortion characteristics.

Table 12. Evaluation on the synthetic Shape Grading dataset. LoCaSE achieves competitive performance and higher rank correlation than SAUCD, demonstrating good generalizability to synthetic distortions.

Method	PLCC	SROCC	KROCC
SAUCD	**0.598**	0.611	0.453
LoCaSE	0.471	**0.651**	**0.473**

6.8. Computational efficiency

We evaluate the computational efficiency of our metric compared to previous metrics in GFLOPs. The number of vertices is set to 10,000, and for IoU, the resolution is set to $256 \times 256 \times 256$. We observe that, as a learnable metric, our metric can achieve good performance while being reasonably computationally efficient. The result is shown in Tab. 13

Table 13. Computational complexity comparison of different metrics.

Metrics	CD	IoU	F-score	P2S	ND	UHD	SAUCD	Ours (Learnable)
GFLOPs	1.6	0.084	2.0	2.0	3.0	0.8	4000	69.94

7. Broader impact and future work

The proposed LoCaSE metric aims to bridge the gap between geometric fidelity and human-perceived realism in 3D shape evaluation. As 3D content becomes increasingly prevalent in gaming, virtual reality, digital humans, and simulation-driven industries, having an evaluation metric aligned with human perception can significantly improve the quality and trustworthiness of generated content. Our metric may assist designers and researchers in producing more perceptually accurate 3D models, reducing trial-and-error cycles and manual inspection. However, there is also potential for misuse, such as optimizing 3D shapes to deceptively maximize perceived realism while compromising physical plausibility or safety (e.g., in medical or robotics applications). We encourage future work to consider fairness, interpretability, and misuse detection in perceptual evaluation metrics.

There are several promising directions for future research. First, while our metric is trained on 3D meshes with human-annotated fidelity, expanding to other modalities (e.g., point clouds, implicit fields, or textured Gaussians) could improve its generality. Second, integrating temporal consistency and motion-awareness may enable perceptual fidelity evaluation for dynamic 3D content such as animations or volumetric videos. Third, further improving robustness under noisy or incomplete input conditions could benefit real-world deployment. Lastly, developing a no-reference metric version that does not rely on a ground truth mesh remains an open and impactful challenge.

References

[1] Artec 3D. Artec 3d - professional 3d scanners. 3, 4

[2] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *ICML*, pages 950–959, 2020. 3

[3] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, pages 2553–2560. IEEE, 2022. 3, 4

[4] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023. 7

[5] Zixuan Huang, Mark Boss, Aaryaman Vasishta, James M Rehg, and Varun Jampani. Spar3d: Stable point-aware re-

construction of 3d objects from single images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16860–16870, 2025. 3

[6] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3

[7] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024. 3

[8] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. 36:22226–22246, 2023. 3

[9] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10072–10083, 2024. 3

[10] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 3

[11] Renderbot. Animal 3d models - by renderbot llc. 3, 4

[12] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Int. Conf. Comput. Vis.*, pages 2304–2314, 2019. 3

[13] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 84–93, 2020. 3

[14] Sketchfab. Sketchfab - the best 3d viewer on the web. 3, 4

[15] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 3

[16] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3

[17] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *Eur. Conf. Comput. Vis.*, pages 1–18. Springer, 2024. 3

[18] Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material, 2025. 3

[19] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 3

[20] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 3

[21] Lizhen Wang, Zhiyua Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. Faceverse: a fine-grained and detail-controllable 3d face morphable model from a hybrid dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2022)*, 2022. 3, 4

[22] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *Eur. Conf. Comput. Vis.*, pages 57–74. Springer, 2024. 3

[23] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[24] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. Icon: Implicit clothed humans obtained from normals. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13286–13296, 2022. 3

[25] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[26] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 3

[27] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. *arXiv preprint arXiv:2405.07510*, 2024. 3

[28] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 3, 4

[29] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5746–5756, 2021. 3, 4

[30] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[31] Hao Zhu, Haotian Yang, Longwei Guo, Yidi Zhang, Yanru Wang, Mingkai Huang, Menghua Wu, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: 3d facial dataset and benchmark for single-view 3d face reconstruction. *TPAMI*, 2023. 3, 4