Architectural Co-LOD Generation (Supplementary Materials)

RUNZE ZHANG, Shenzhen University, China
SHANSHAN PAN, Shenzhen University, China
CHENLEI LV, Shenzhen University, China
MINGLUN GONG, University of Guelph, Canada
HUI HUANG*, Shenzhen University, China

CCS Concepts: • **Computing methodologies → Shape inference**.

Additional Key Words and Phrases: 3D generation, structural reconstruction, level of detail, co-analysis, architectural models

ACM Reference Format:
Runze Zhang, Shanshan Pan, Chenlei Lv, Minglun Gong, and Hui Huang. 2024. Architectural Co-LOD Generation (Supplementary Materials). *ACM Trans. Graph.* 43, 6, Article 1 (December 2024), 40 pages. https://doi.org/10.1145/3687905

The supplementary materials provide additional technical details about Co-LOD, including D2 descriptor computation (Sec. 1), Co-LOD for Single Building (Sec. 2), and BSP implementation (Sec. 3). Additionally, we outline the configurations of comparative methods and present more comprehensive results (Sec. 4).

1 CALCULATION OF D2 DESCRIPTOR

Considering computation time, robustness, and fidelity, we opted for the D2 descriptor [Osada et al. 2002] for shape measurement over more advanced deep shape descriptor solutions [Xie et al. 2017]. In order to compute the D2 descriptor for a given segment efficiently and reliably, we employ a specific pipeline outlined in Algorithm 1. Initially, we uniformly sample 10,000 points from the segment and apply anisotropic scaling to normalize them into a point cloud p_n [Kazhdan et al. 2004], ensuring consistent shape representation. Subsequently, we randomly select 256×256 point pairs from p_n and calculate their distances, dividing them into 256 intervals for quantification. This process yields the initial 256 components of the D2 vector. Furthermore, we sample 10,000 normal vectors from the segment, randomly selecting 20,000 pairs and computing the angle between each pair. These angles are discretized into 10 bins within the [0, 180] degree range, generating the remaining 10 components of the D2 vector.

*Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Authors' addresses: Runze Zhang, oliverzrz.cyber@gmail.com, Shenzhen University, China; Shanshan Pan, psshappystar@gmail.com, Shenzhen University, China; Chenlei Lv, lvchenlei@gmail.com, Shenzhen University, China; Minglun Gong, minglun@uoguelph.ca, University of Guelph, Canada; Hui Huang, hhzhiyan@gmail.com, College of Computer Science & Software Engineering, Shenzhen University, China.

0730-0301/2024/12-ART1 $15.00
https://doi.org/10.1145/3687905

ALGORITHM 1: Calculation of D2 Descriptor

Input: Segment
Output: D2 Descriptor
// Sample points and apply transformations
$sample\_points \leftarrow$ uniform_sampling($segment$, 10000)
$normalized\_cloud \leftarrow$ apply_transformations($sample\_points$)
// Compute distance-based features
$dis\_histogram \leftarrow$ compute_dis_histogram($normalized\_cloud$)
// Compute angle-based features
$angle\_histogram \leftarrow$ compute_angle_histogram($segment$)
// Combine distance and angle histograms to form D2 vector
return $concatenate(distance\_histogram, angle\_histogram)$

2 CO-LOD FOR SINGLE BUILDING

To conduct database-based single building analysis, we need to establish a component database and convert the LOD hierarchical method designed for co-analysis into a form that allows analyzing individual buildings. We'll explain these two steps and present the results of our single-building analysis.

Database Construction. We manually selected 90 representative structural segments from *Composite Scene* that represent building components outside the primary architectural structure. These segments enable us to hierarchically classify the components within individual buildings.

Method. Compared to the original Co-LOD algorithm, we adapt the LOD hierarchy derived from co-analysis to a format suitable for data-driven examination while maintaining other algorithmic components. For a building model divided into structural segments, we calculate $f_r(s)$ and $f_s(s)$ using the same approach as in co-analysis:

$$f_r(s) = \mathrm{IoU}(s, I) = \mathrm{IoU}(P(s), P(I)), \tag{1}$$

$$f_s(s) = -\frac{\mathrm{area}(s)}{\mathrm{area}(I)}, \tag{2}$$

where $P(X)$ represents the discrete sampled point set of structural segments X, and I represents all segments of the model. To incorporate the structural segments from the database into our analysis, we introduce a new term, $f'_{co}(s)$, replacing the original f_{co}. This new term is defined as:

$$f'_{co}(s) = \sum_{s' \in S_d} e^{-\mathrm{dis}(s, s')}, \tag{3}$$

where s is a segment, and S_d represents the collection of structural segments in the database. These three terms substitute the optimization equation in co-analysis to determine whether a segment

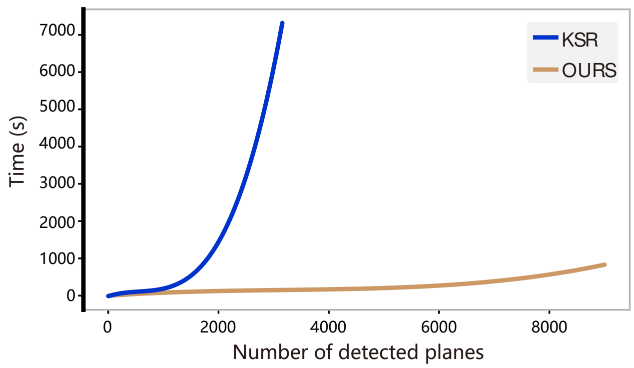

Fig. 1. Relationship between the runtime of the Polygonal Mesh Extraction stage and the number of detected planes. Compared to KSR, our method efficiently handles a greater number of input planes.

belongs to LOD0. Segments exhibiting a positive weighted sum are classified as LOD0, where the weights assigned to each item are consistent with those defined during co-analysis. Remaining segments are then clustered and assigned to higher LOD levels.

This method effectively achieves database-based LOD generation for individual buildings, as tested on *Campus* and *European City*, with statistical results shown in Table 1 and visual results in Fig. 2. Co-LOD for single building proved more effective in datasets similar to the database (*Campus*). However, the performance on datasets beyond the database scope (*European City*) was significantly inferior to that based on co-analysis. This is reflected in the higher complexity of the generated LOD0 models and the lack of corresponding improvements in reconstruction accuracy.

3 POLYGONAL MESH EXTRACTION - BSP

We provide the pseudocode for the spatial partitioning strategy outlined in the Polygonal Mesh Extraction section (refer to Algorithm 2). Our Polygonal Mesh Extraction approach is more efficient than KSR, as demonstrated by the runtime comparison in Fig. 1, where the KSR implementation is sourced from CGAL 6.0.

4 COMPARATIVE EXPERIMENT SETTINGS AND ADDITIONAL RESULTS

The experimental setup for the comparison methods PolyFit, KSR, LowPoly, QEM, and RobustLowPoly is outlined as follows. For PolyFit, the plane detection parameters were configured with a maximum distance ($d_{\max}$) of 0.5, a minimum region size ($s_{\min}$) of 1000, and a maximum angle ($\theta_{\max}$) of 30 degrees. For KSR, we empirically established three parameter configurations to generate models with varying levels of detail, defined by tuples: $d_{\max}$ at 0.5, 0.3, and 0.1, $s_{\min}$ at 2000, 500, and 100, and $\theta_{\max}$ consistently set at 30 degrees across all settings. For LowPoly, we used the results produced by the official executable. Since other methods automatically generate their results, we utilized the automatically generated Carved Mesh as the final result without manually selecting from the provided pareto curve. Additionally, if results are not produced within an hour, we replace the original input with our highest LOD model with fewer facets to ensure effective output within a reasonable time frame. For both QEM and RobustLowPoly, we reduce the model faces to match the corresponding LOD for fair comparisons.

In Fig. 3, we further demonstrate the results obtained on *Composite Scene*. Figs. 4-33 show the randomly selected individual models used in the user study, and Figs. 34-39 display the scenes employed in the user study. For individual models, we restricted our selection to models with over 1,000 facets to ensure diversity.

ALGORITHM 2: Binary Space Partition

Input: A set of planes, denoted as P
Output: Partitioned space defined by subspaces

Compute the bounding box (bbox) containing all planes in P
Create an empty queue of spaces to be partitioned, q_s
// Enqueue the initial bounding box to the partition queue
q_s.push(bbox)
while *q_s is not empty* **do**
    // Dequeue the next space to be partitioned
    Remove and retrieve the next space S_p from q_s
    Extract the largest plane p from S_p for partitioning
    Divide S_p into two subspaces S_p^1 and S_p^2 using p
    // Distribute remaining planes to their respective subspaces
    for *each plane p' in S_p* **do**
        if *p' belongs to S_p^1* **then**
            Assign p' to S_p^1
        end
        if *p' belongs to S_p^2* **then**
            Assign p' to S_p^2
        end
    end
    // Enqueue non-empty subspaces back to the partition queue
    if *S_p^1 contains planes* **then**
        q_s.push(S_p^1)
    end
    if *S_p^2 contains planes* **then**
        q_s.push(S_p^2)
    end
end

Table 1. Statistics of Co-LOD in different LOD0 generation tasks. S: single building analysis, Co: co-analysis based on the scene.

Task	#F	#V	HD	LFD
S: *Campus*	421	198	0.023	3633
Co: *Campus*	346	179	0.021	3656
S: *European City*	1650	827	0.010	3499
Co: *European City*	1196	601	0.009	3494

REFERENCES

Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. 2004. Shape Matching and Anisotropy. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 23, 3 (2004), 623–629.
Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. 2002. Shape Distributions. *ACM Trans. on Graphics* 21, 4 (2002), 807–832.
Jin Xie, Guoxian Dai, Fan Zhu, Edward K. Wong, and Yi Fang. 2017. DeepShape: Deep-Learned Shape Descriptor for 3D Shape Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 7 (2017), 1335–1345.

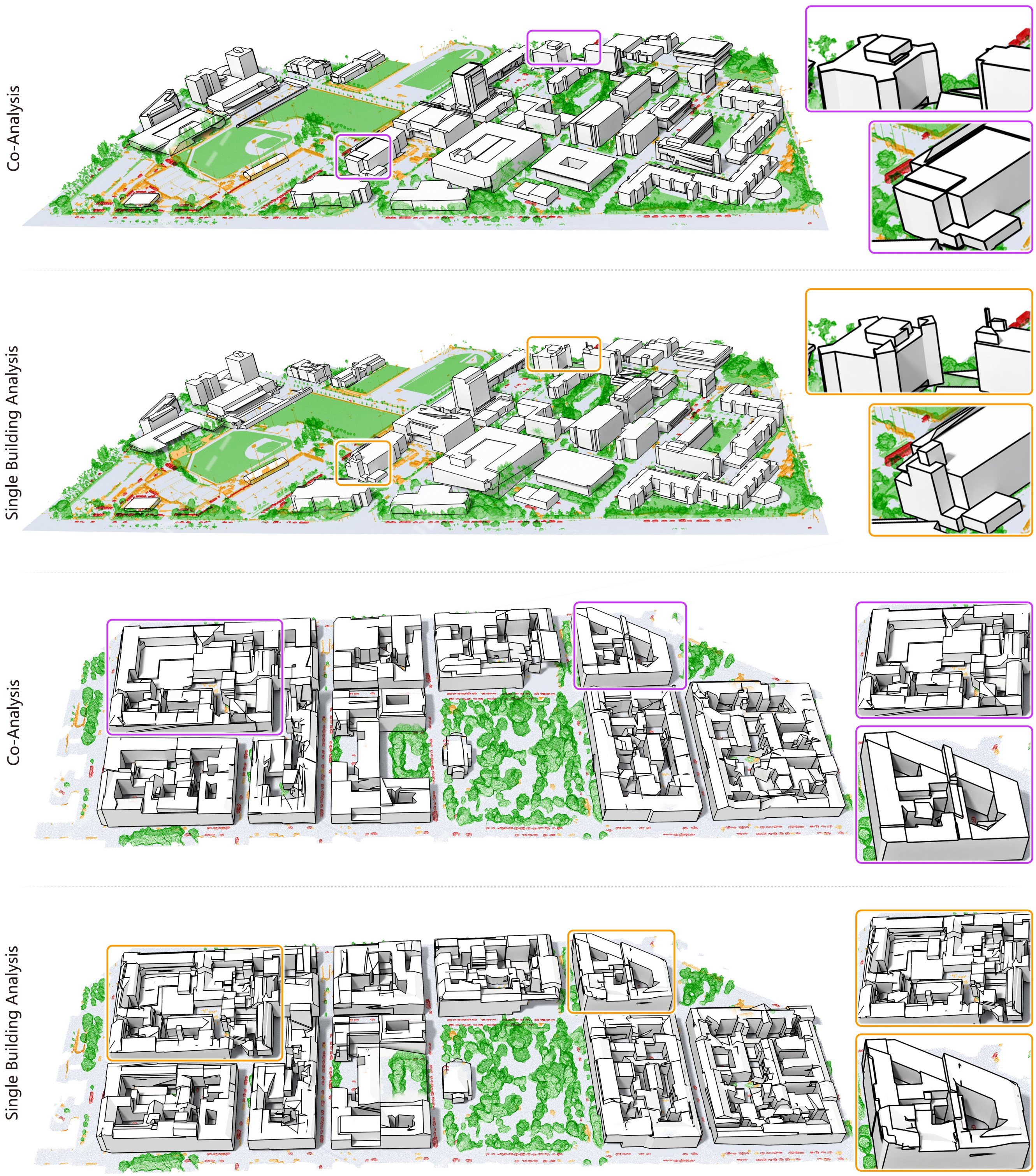

Fig. 2. Comparison of Co-analysis and Co-LOD for Single Building Analysis on *Campus* (top) and *European City* (bottom).

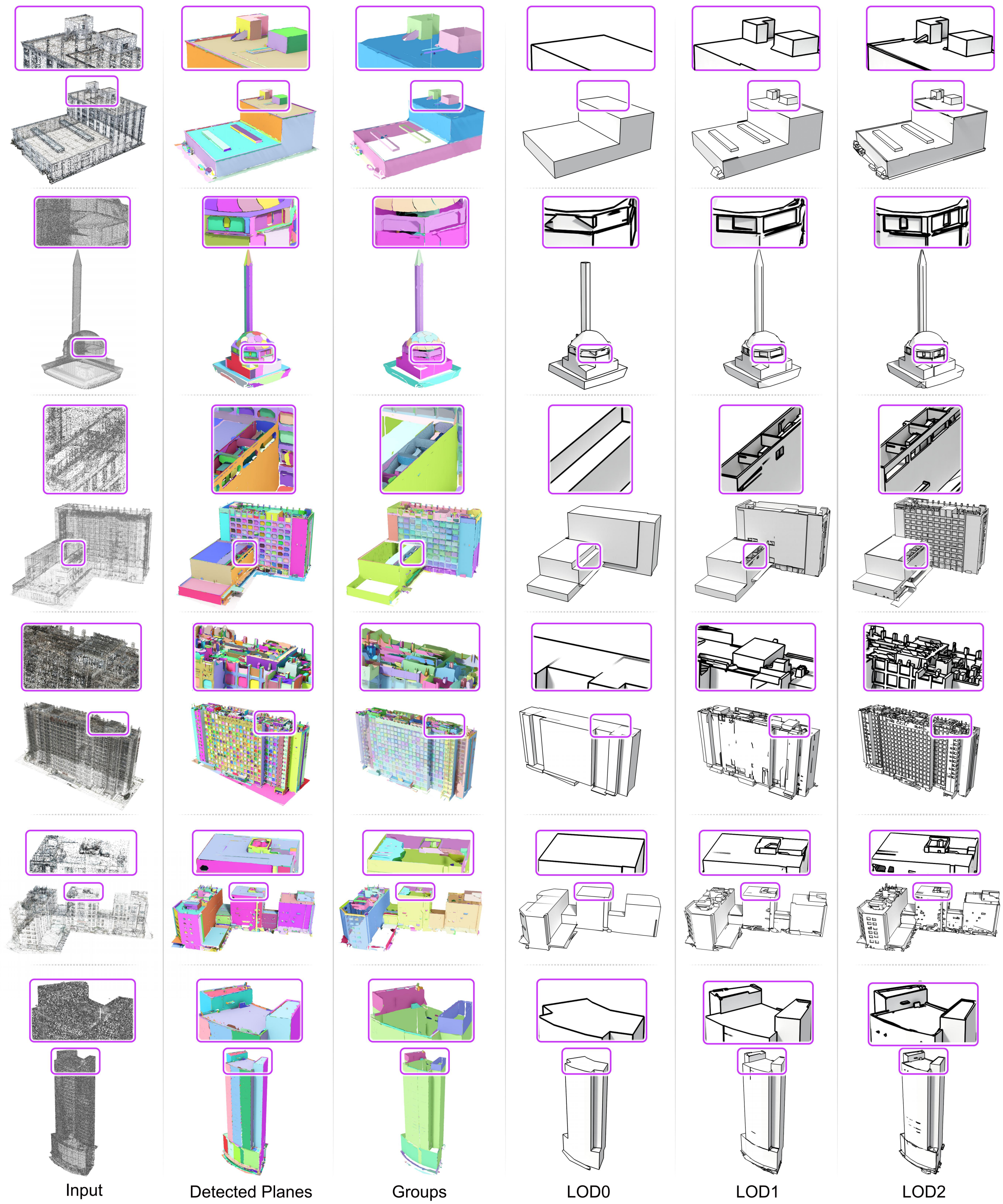

Fig. 3. Further LOD generation results by Co-LOD for *Composite Scene*.

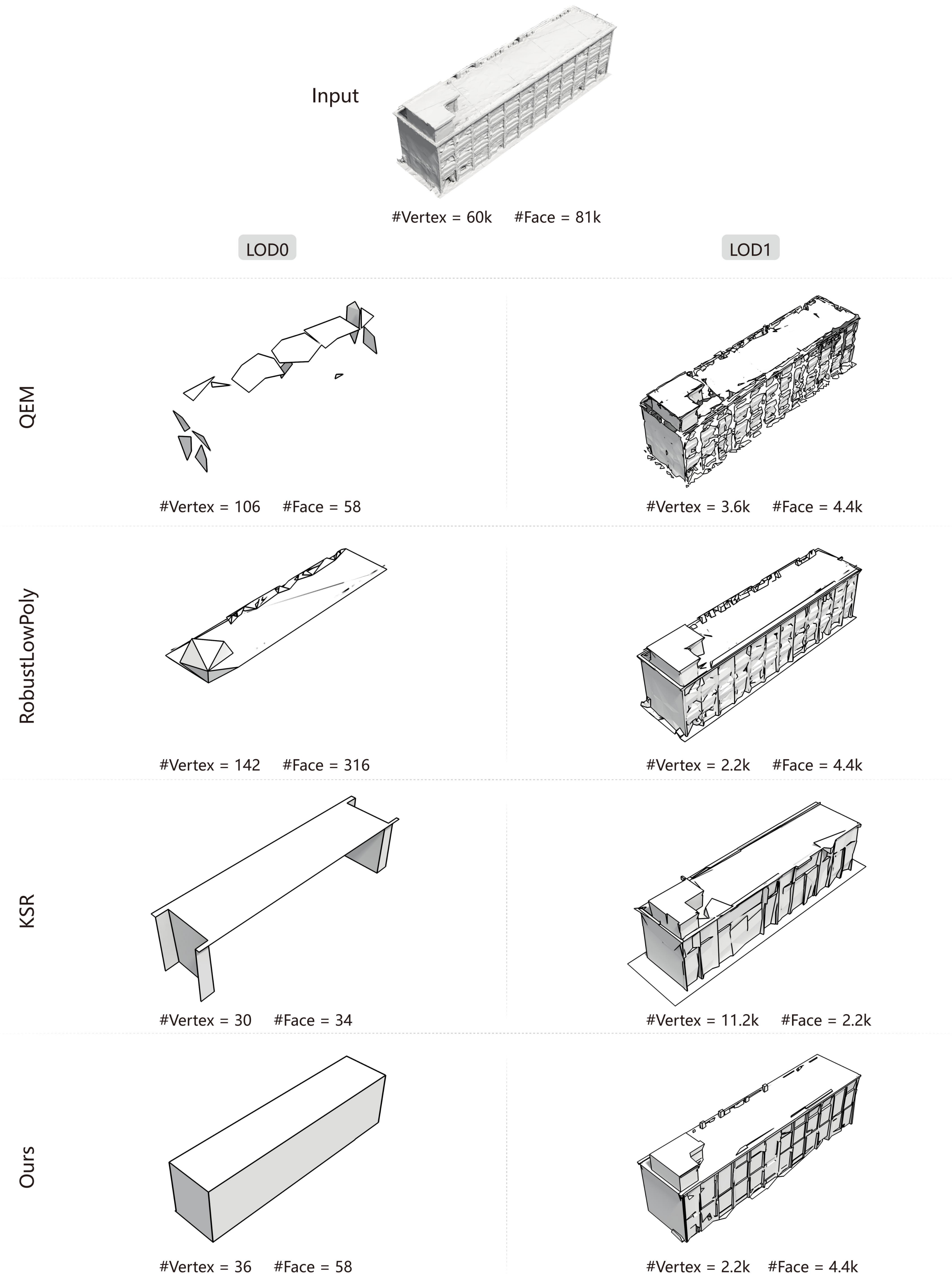

Fig. 4. Qualitative comparisons of randomly selected cases (1/30).

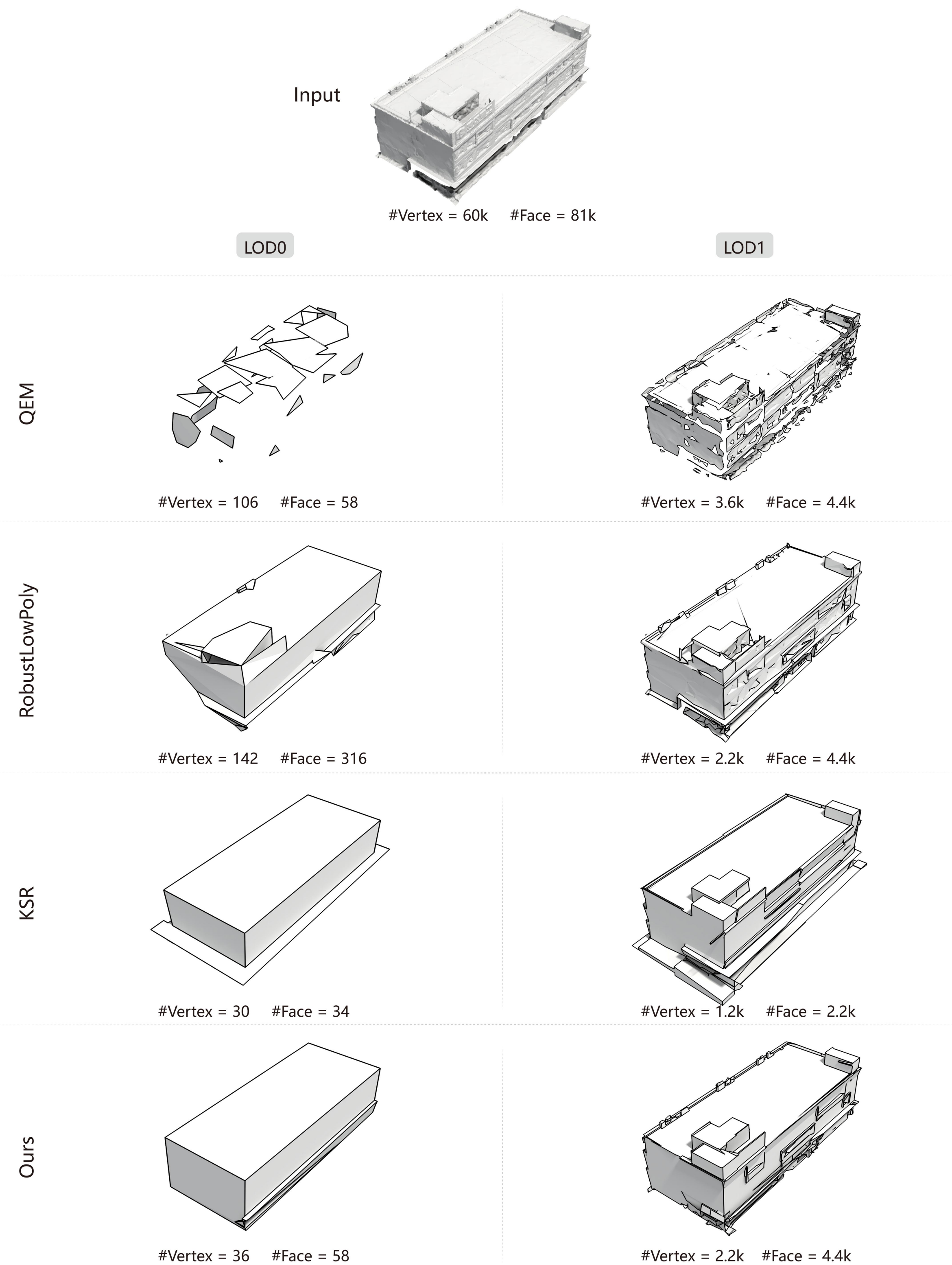

Fig. 5. Qualitative comparisons of randomly selected cases (2/30).

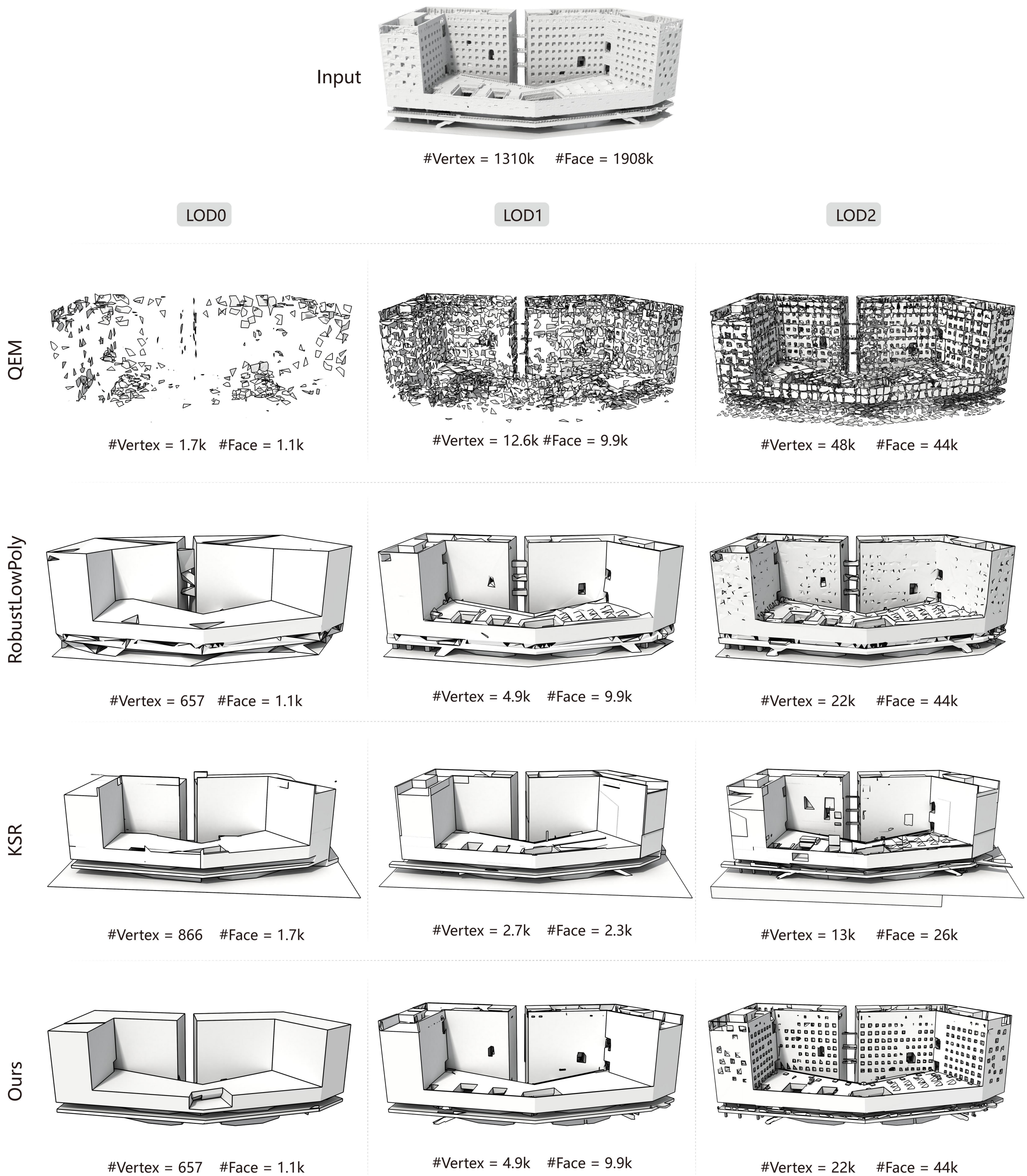

Fig. 6. Qualitative comparisons of randomly selected cases (3/30).

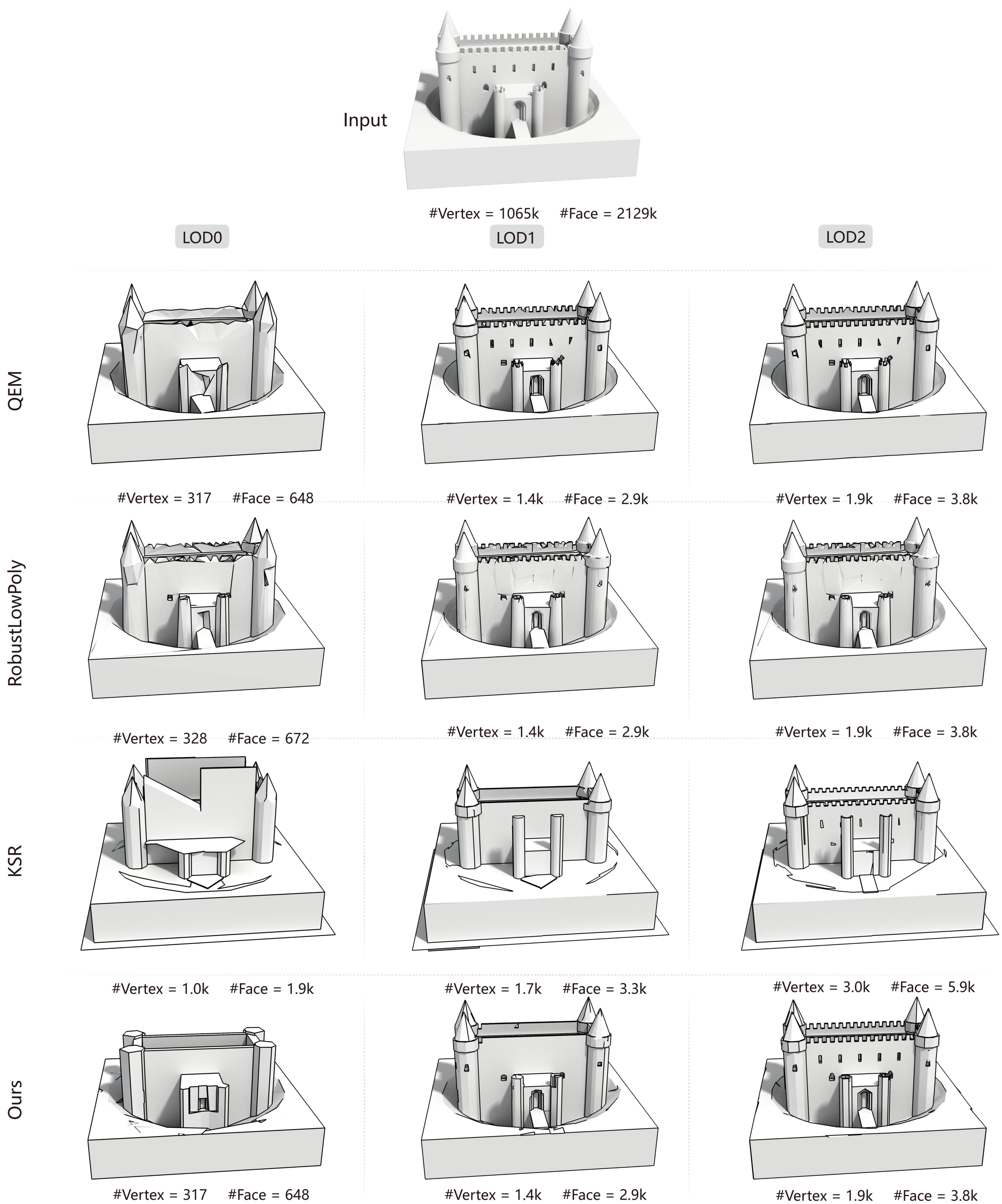

Fig. 7. Qualitative comparisons of randomly selected cases (4/30).

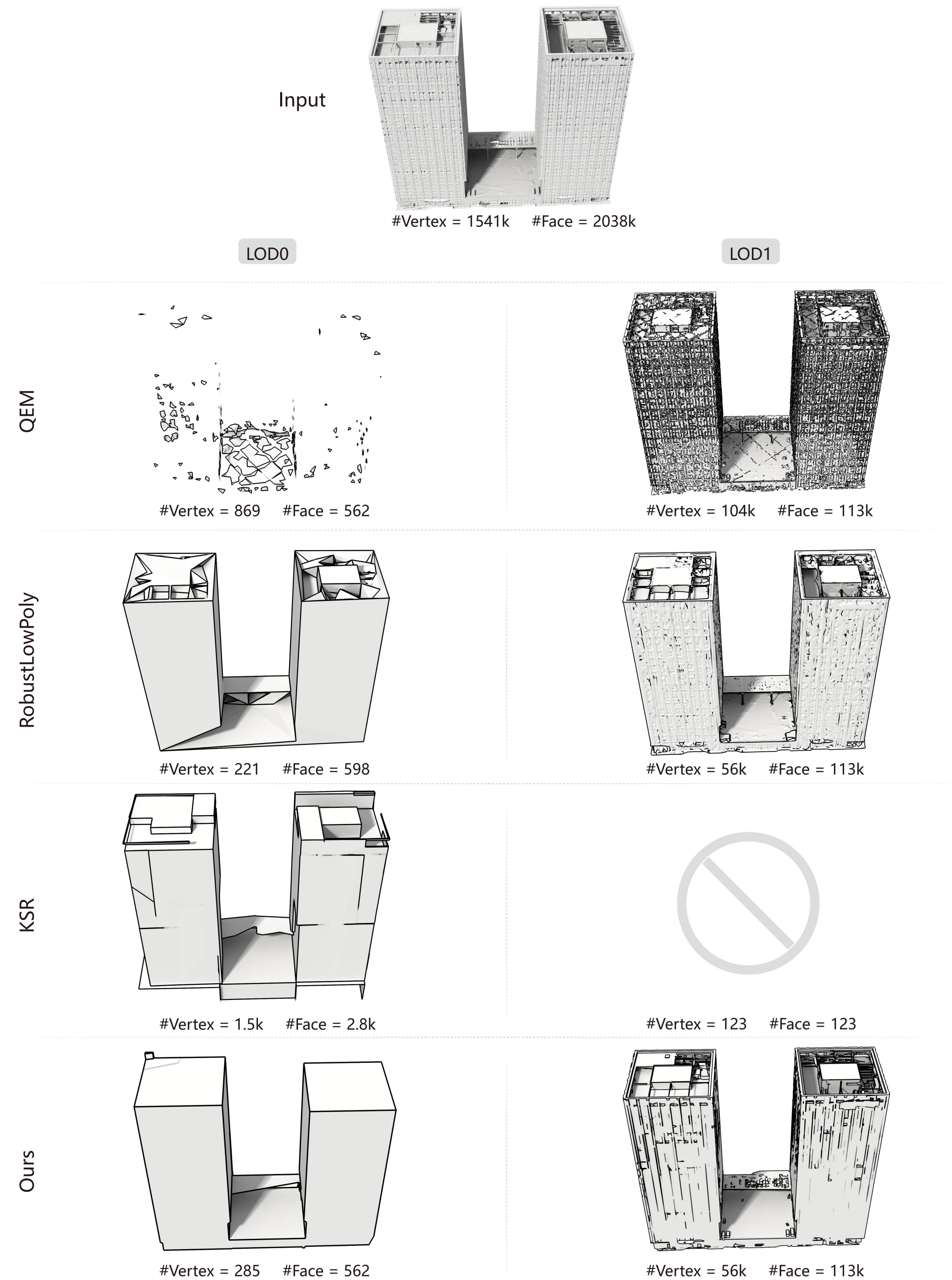

Fig. 8. Qualitative comparisons of randomly selected cases (5/30).

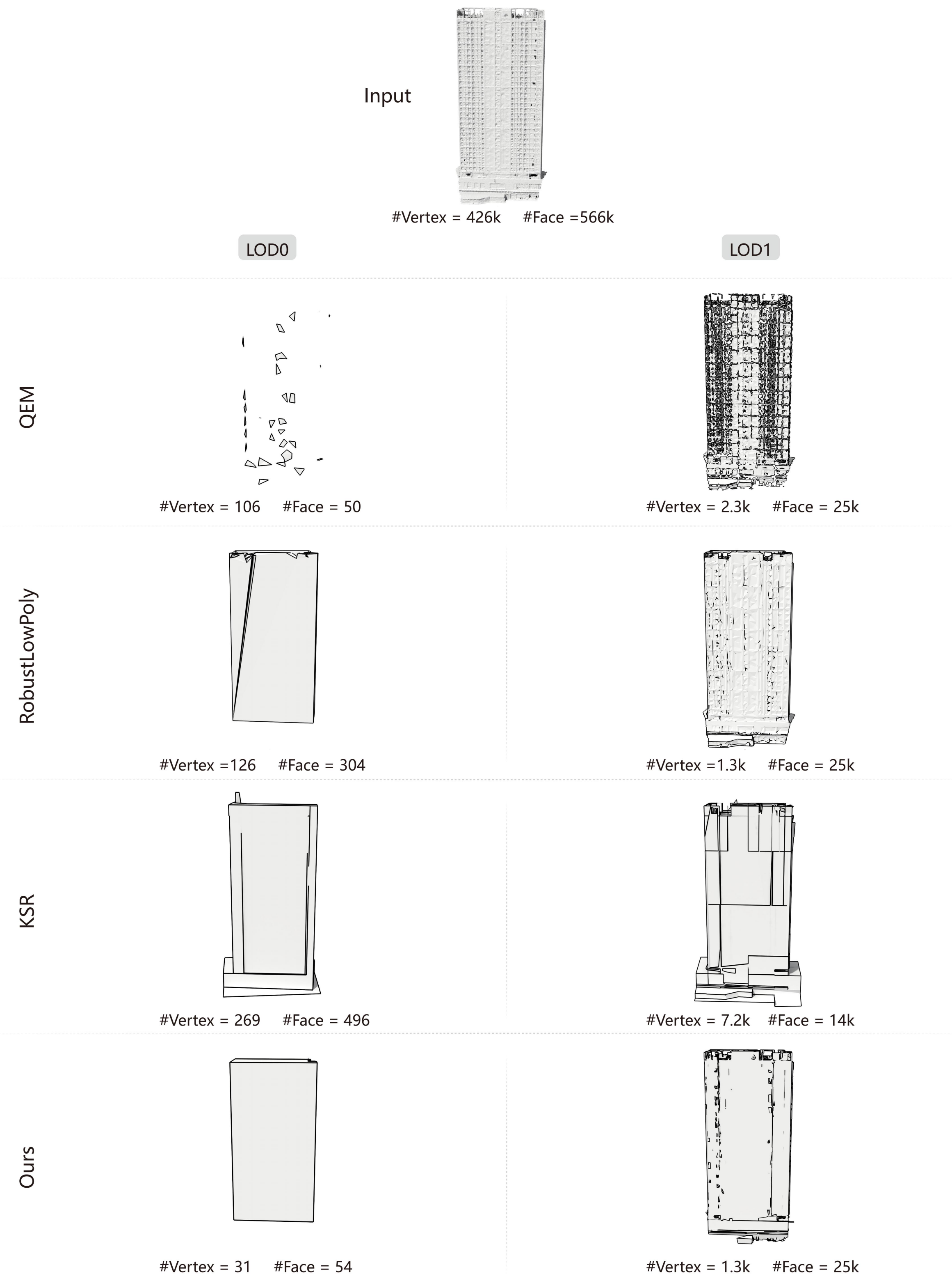

Fig. 9. Qualitative comparisons of randomly selected cases (6/30).

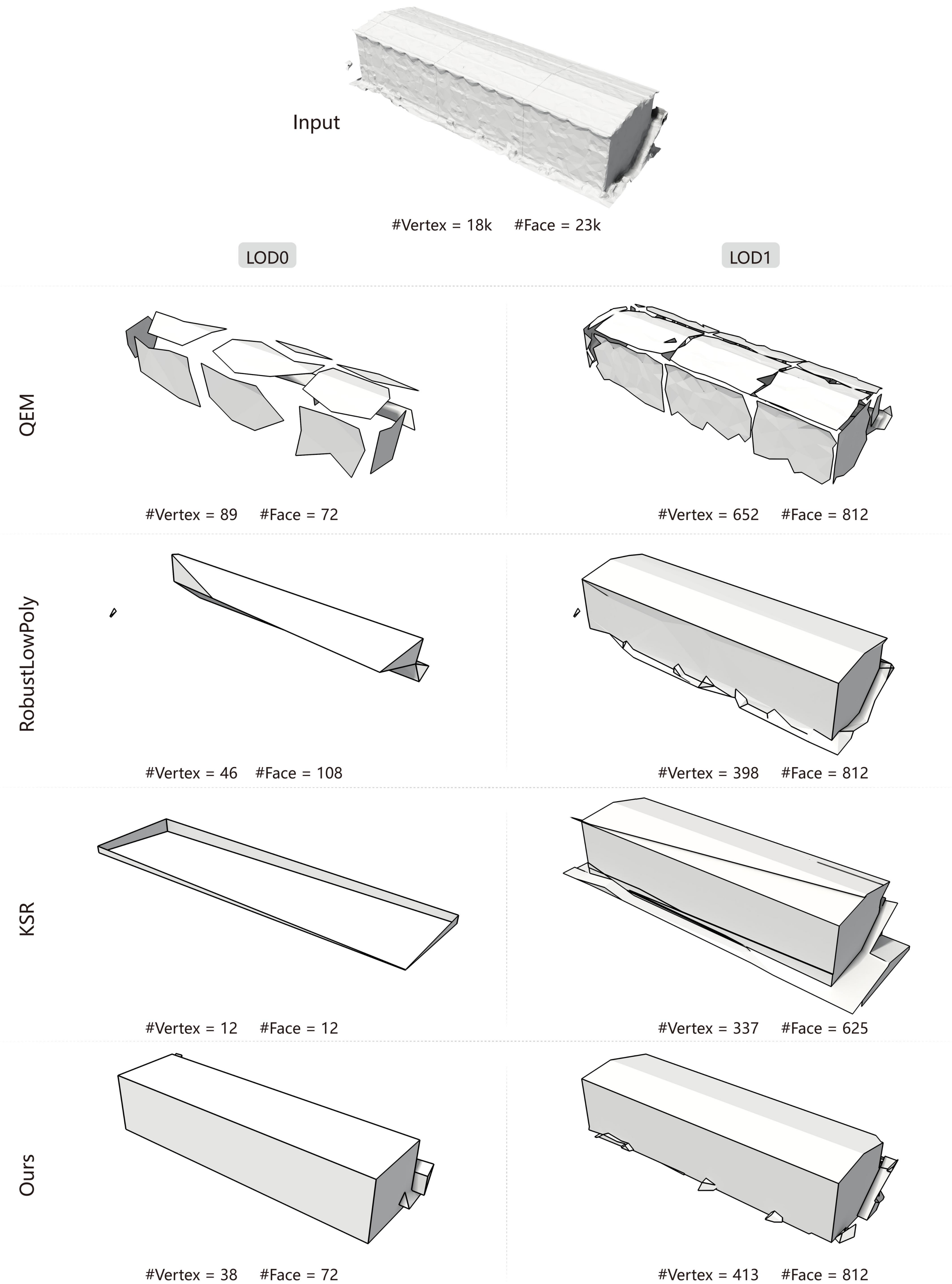

Fig. 10. Qualitative comparisons of randomly selected cases (7/30).

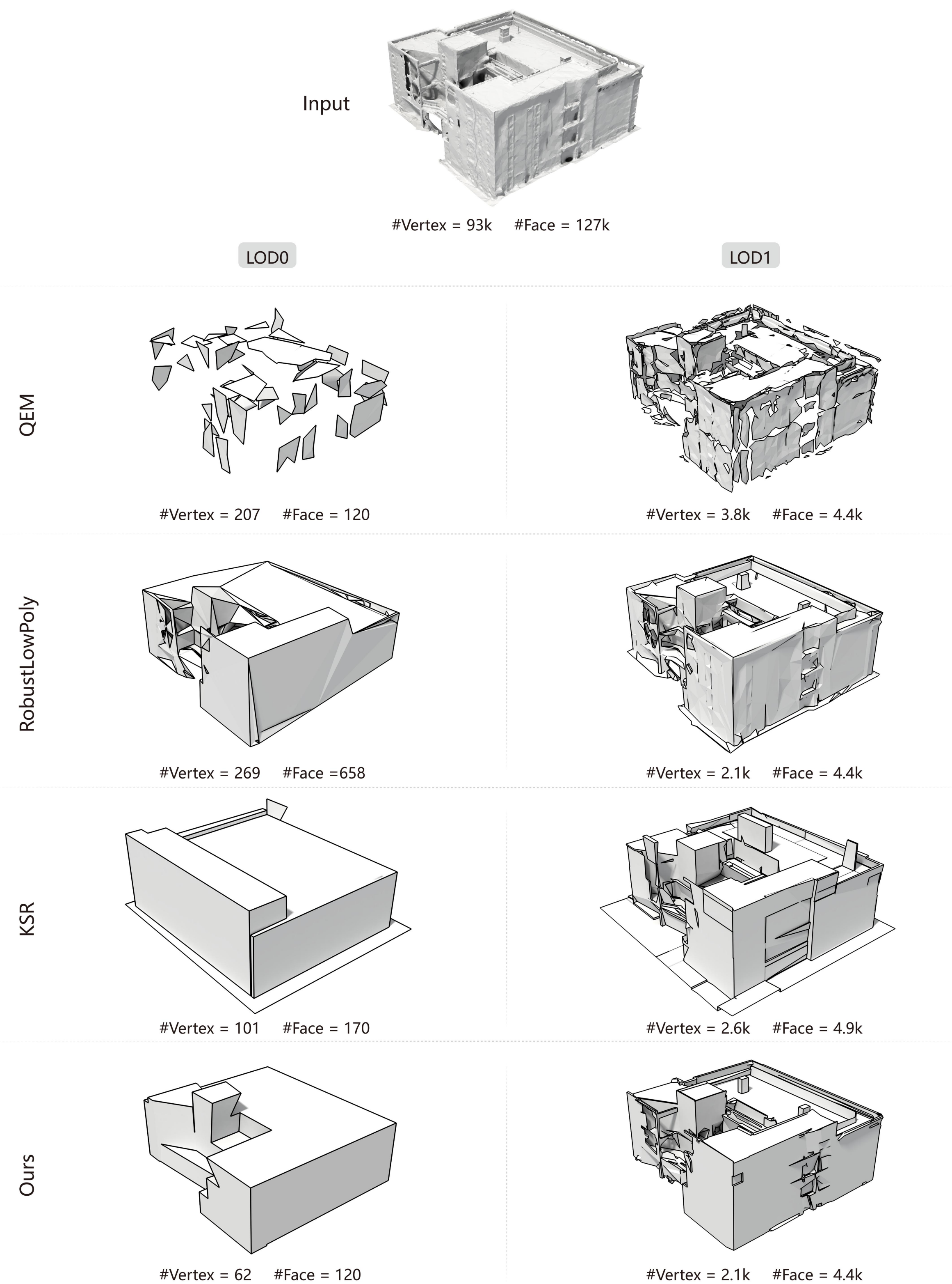

Fig. 11. Qualitative comparisons of randomly selected cases (8/30).

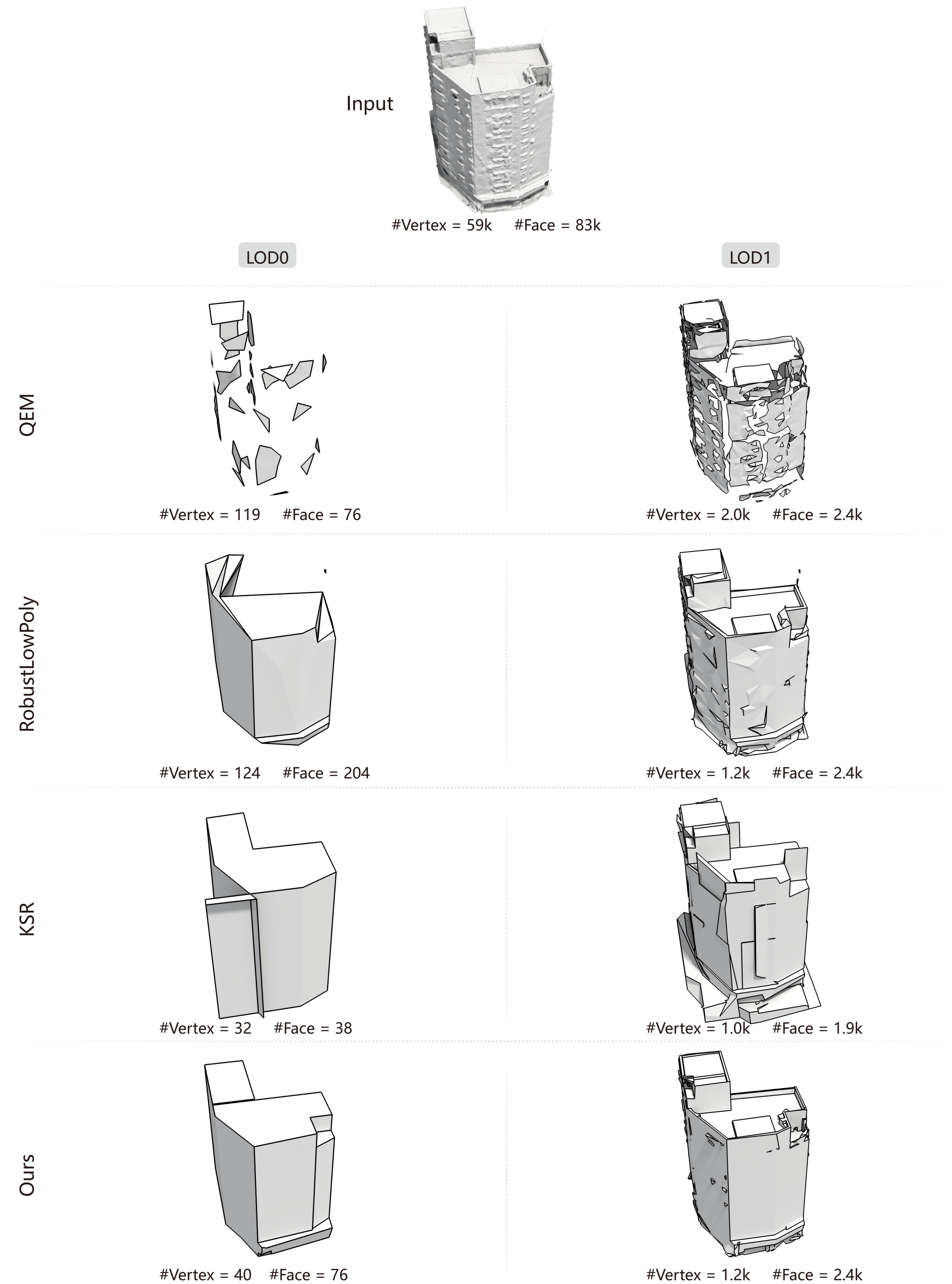

Fig. 12. Qualitative comparisons of randomly selected cases (9/30).

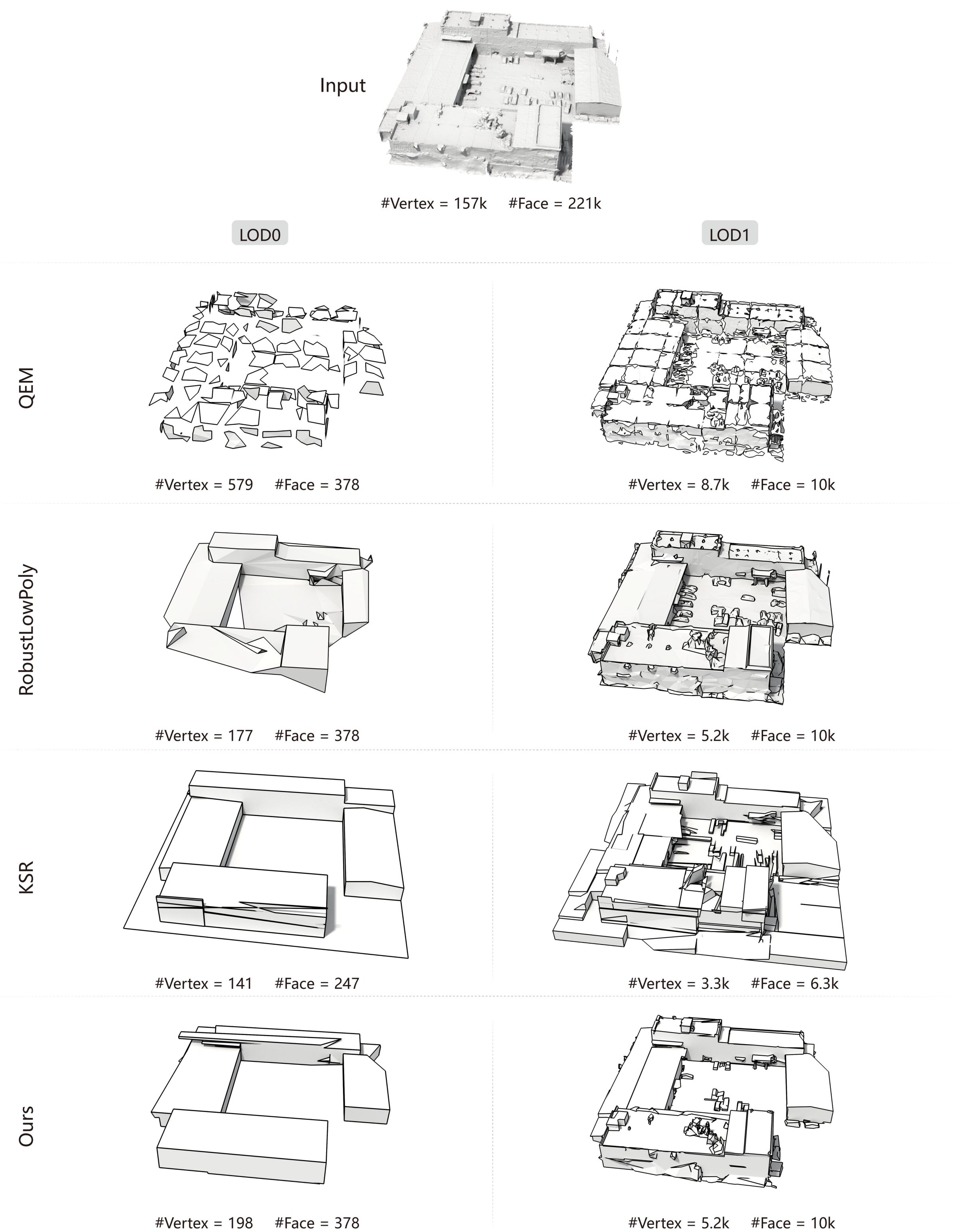

Fig. 13. Qualitative comparisons of randomly selected cases (10/30).

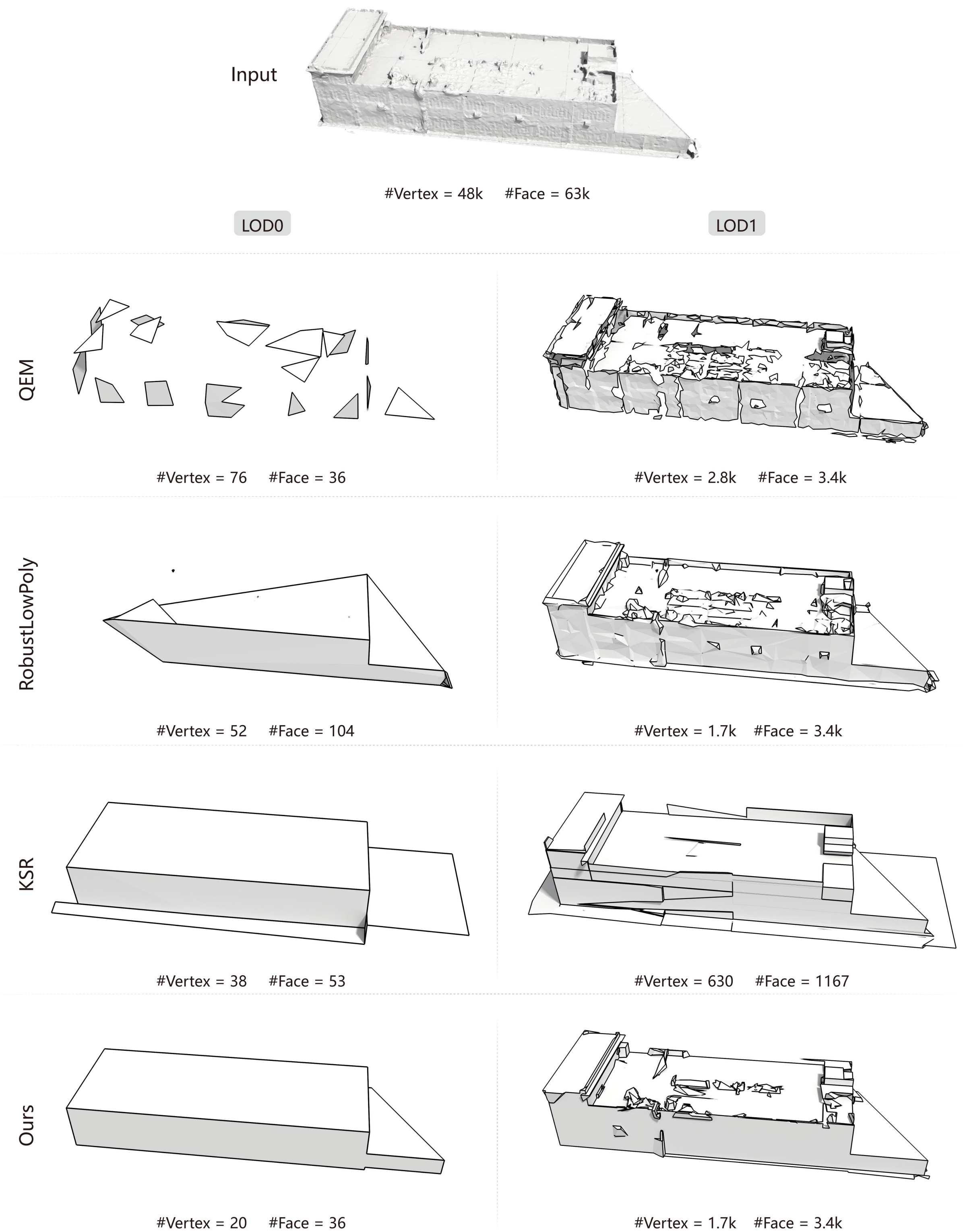

Fig. 14. Qualitative comparisons of randomly selected cases (11/30).

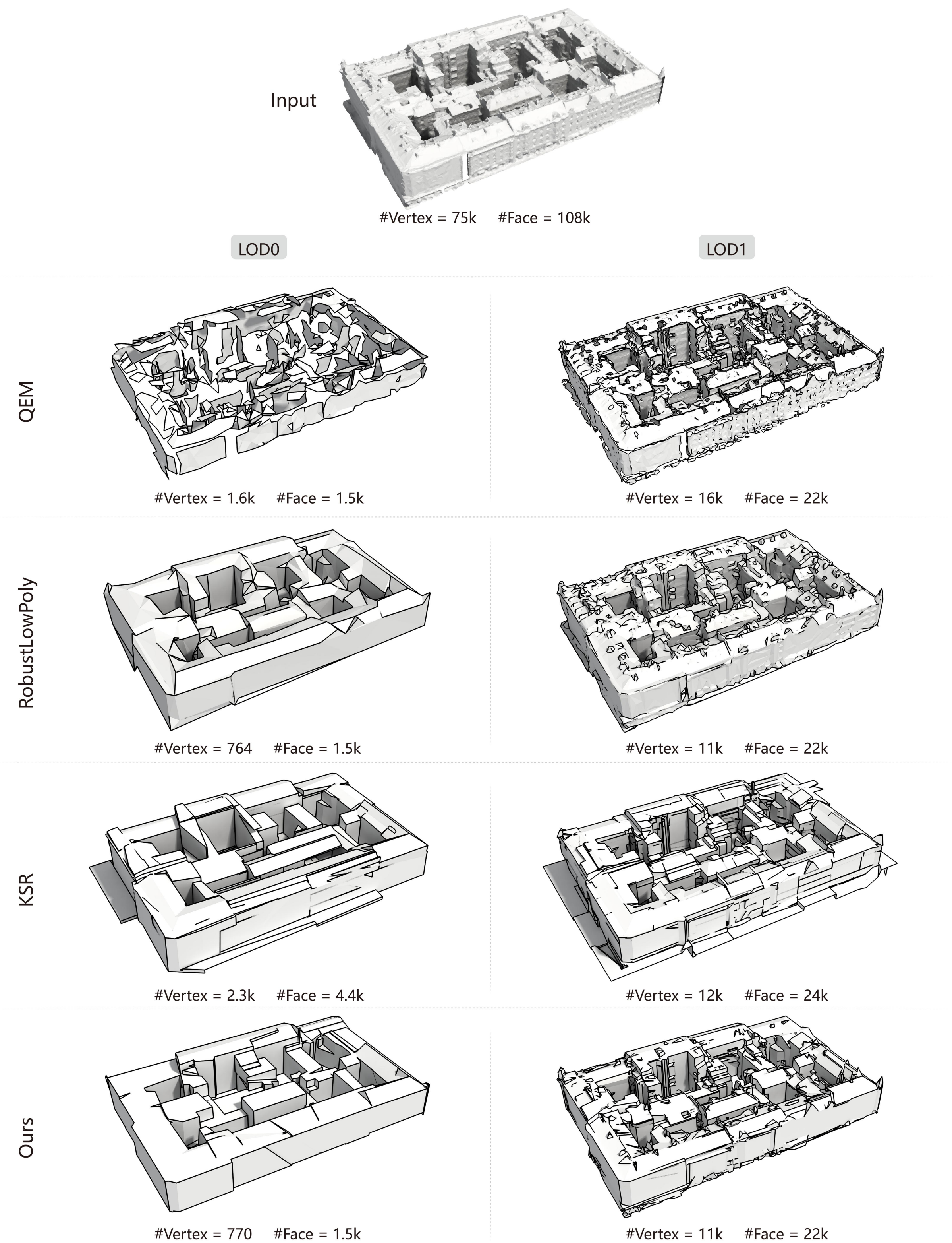

Fig. 15. Qualitative comparisons of randomly selected cases (12/30).

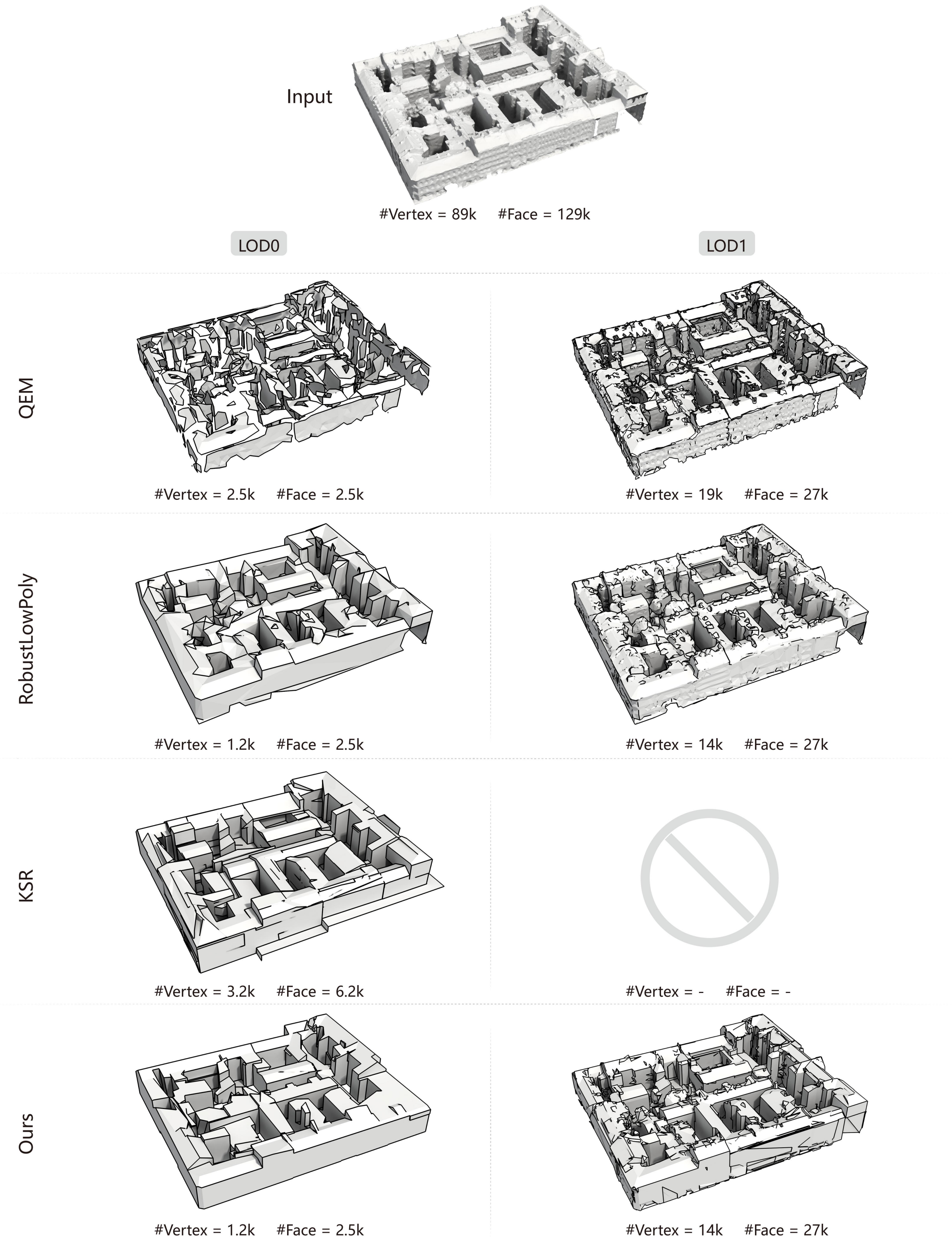

Fig. 16. Qualitative comparisons of randomly selected cases (13/30).

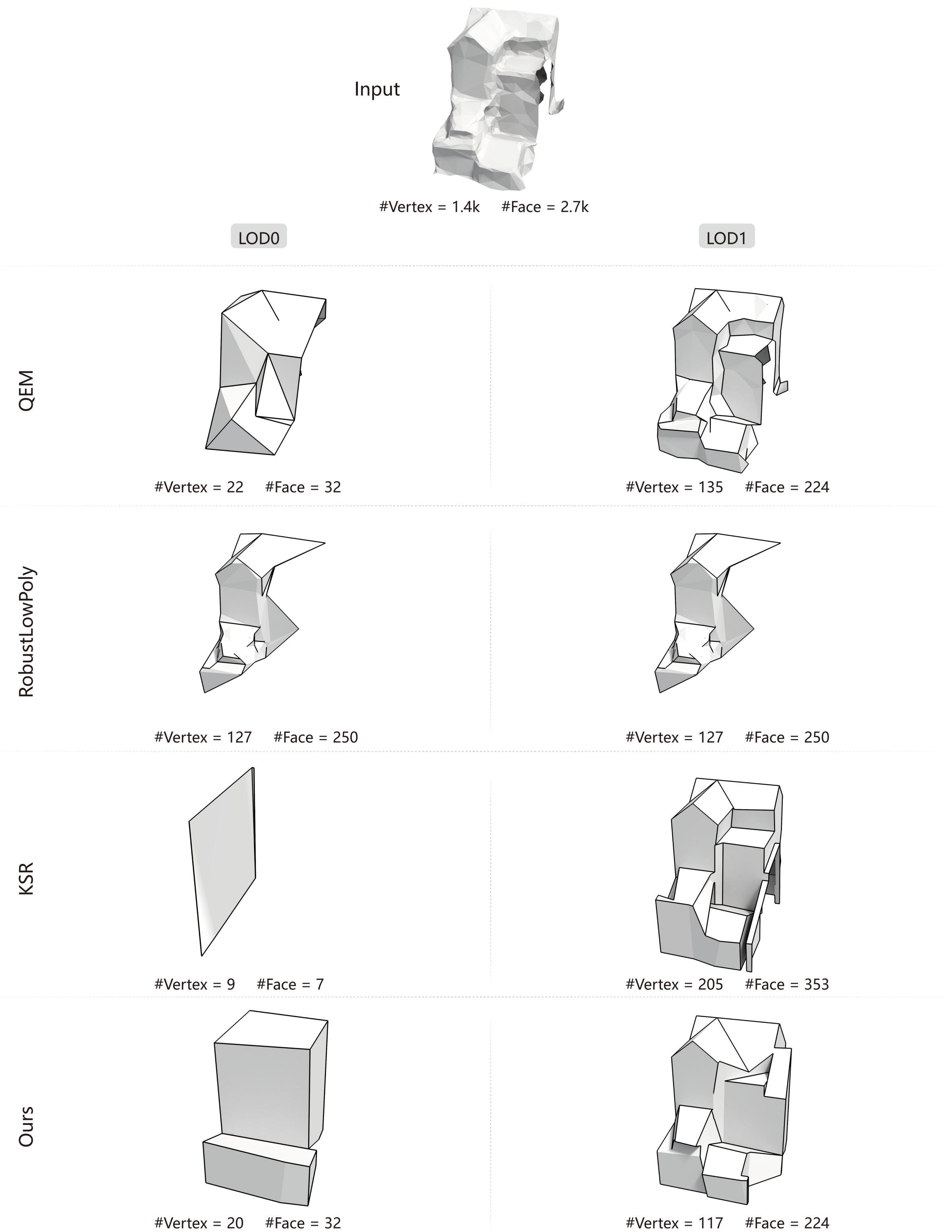

Fig. 17. Qualitative comparisons of randomly selected cases (14/30).

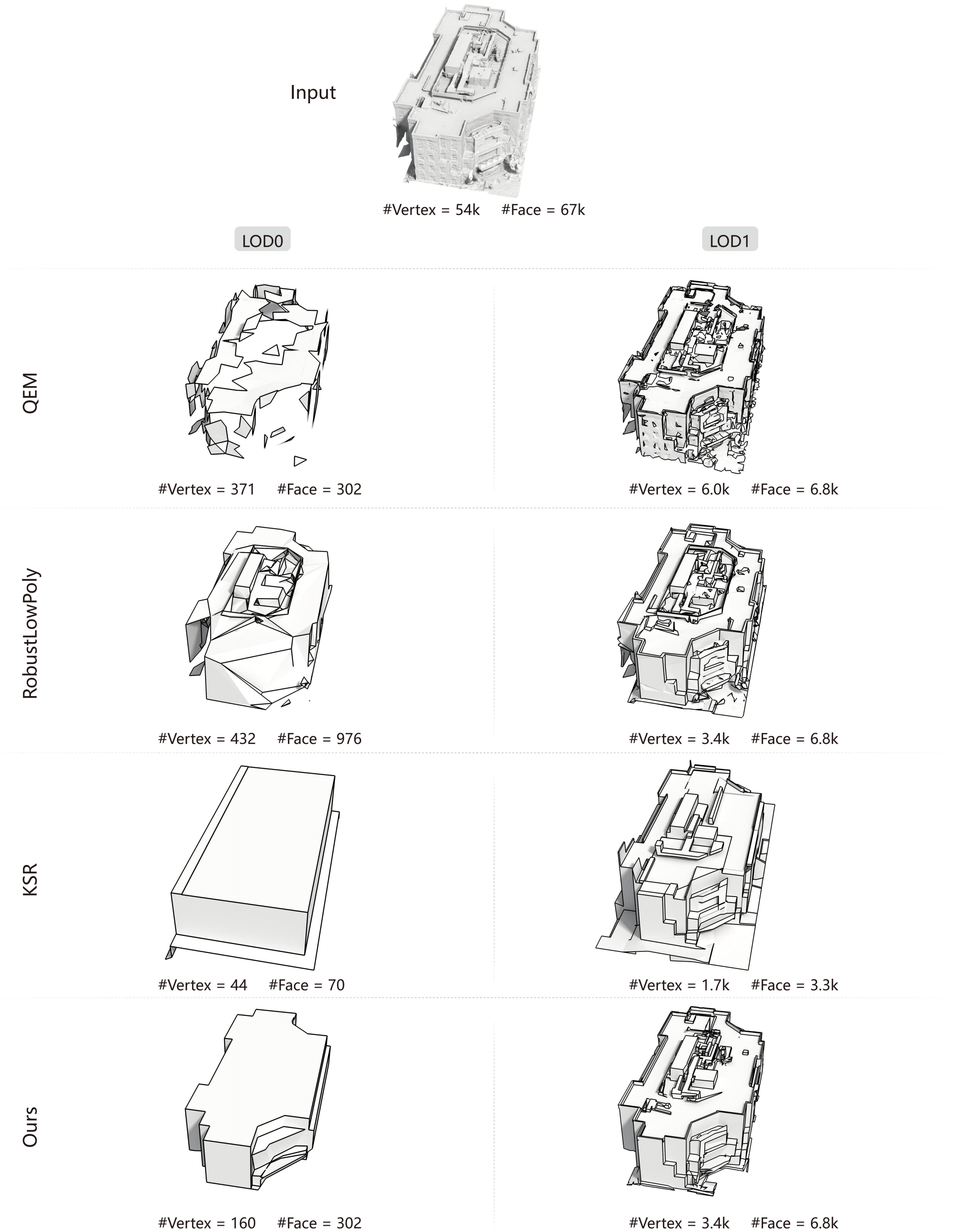

Fig. 18. Qualitative comparisons of randomly selected cases (15/30).

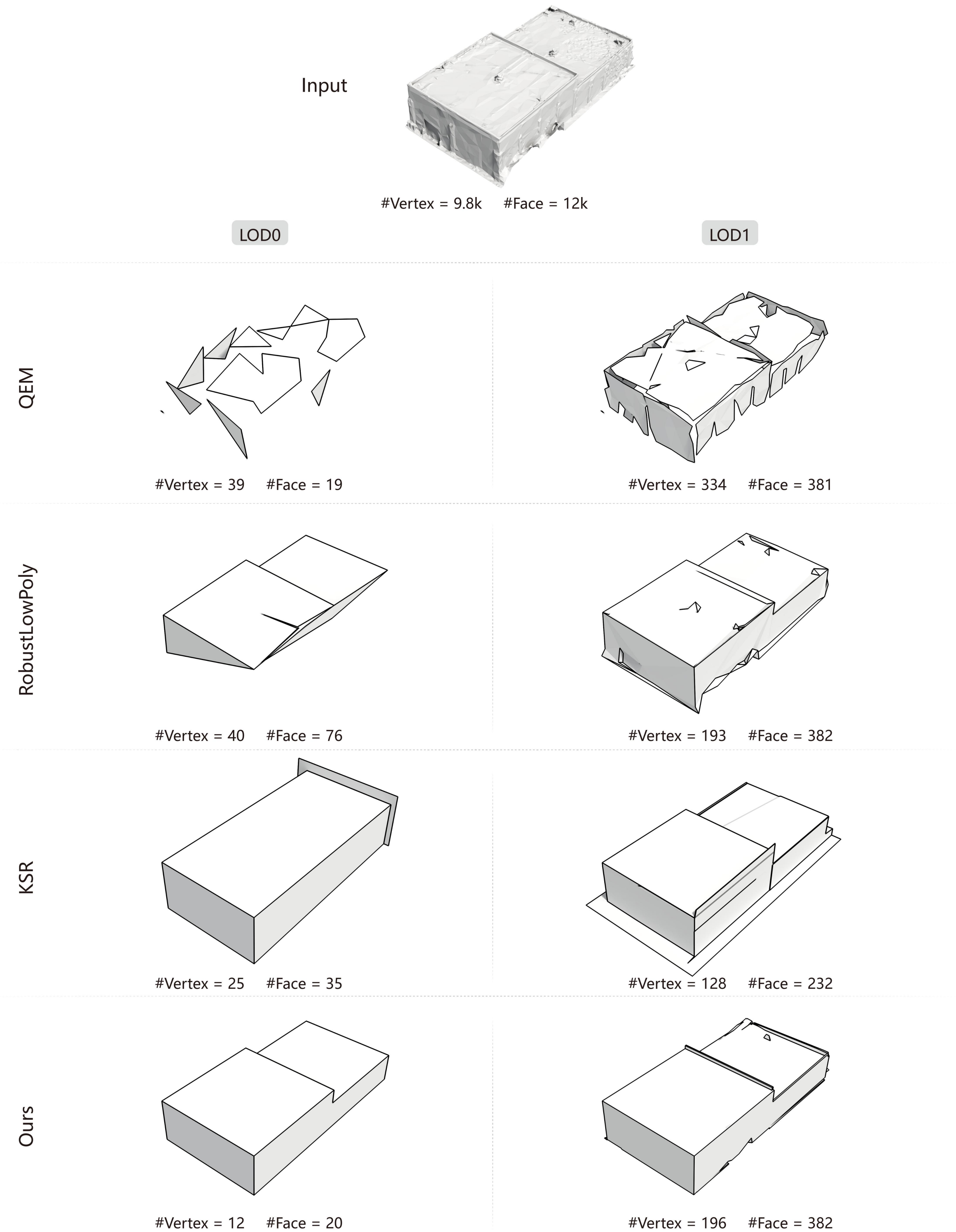

Fig. 19. Qualitative comparisons of randomly selected cases (16/30).

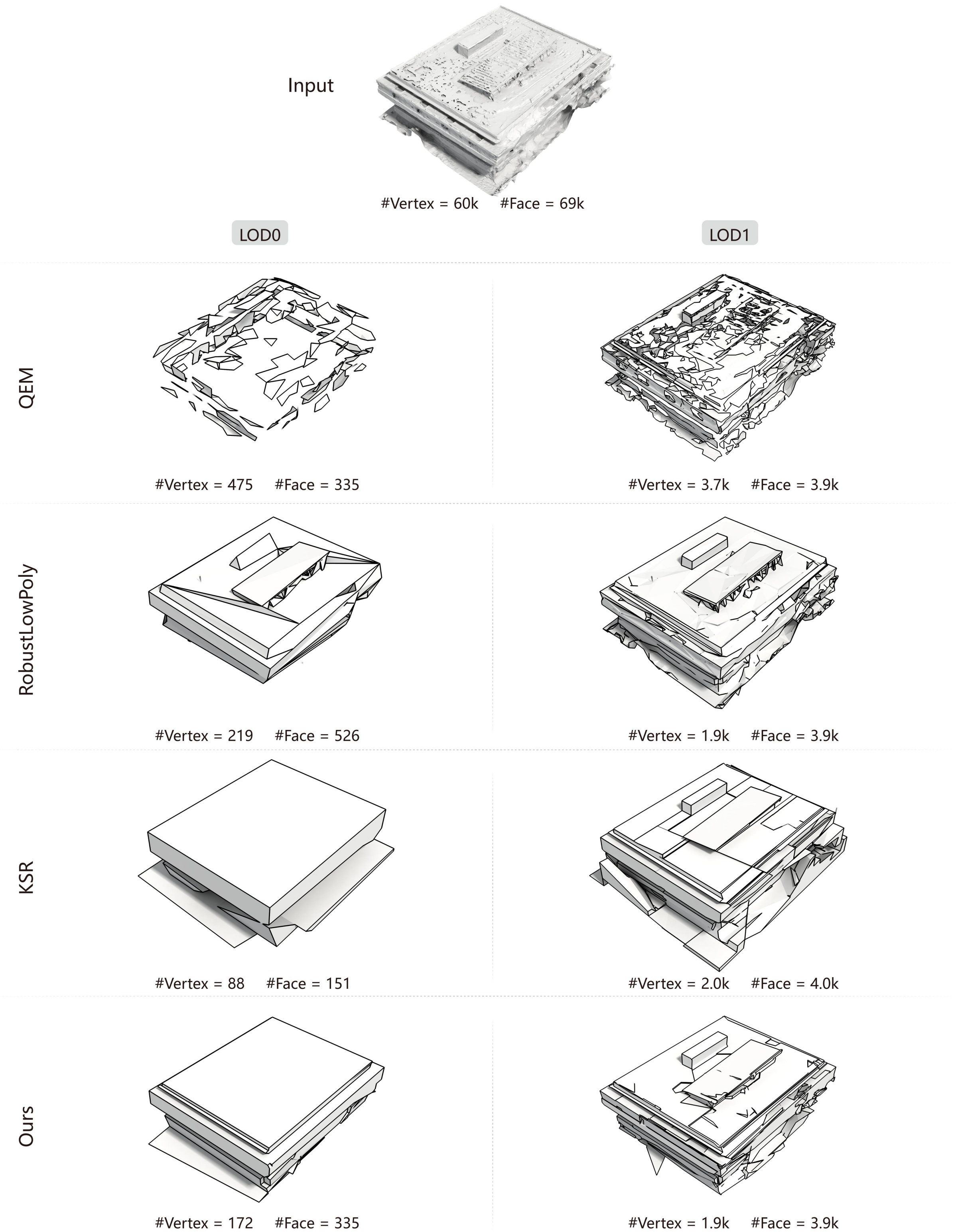

Fig. 20. Qualitative comparisons of randomly selected cases (17/30).

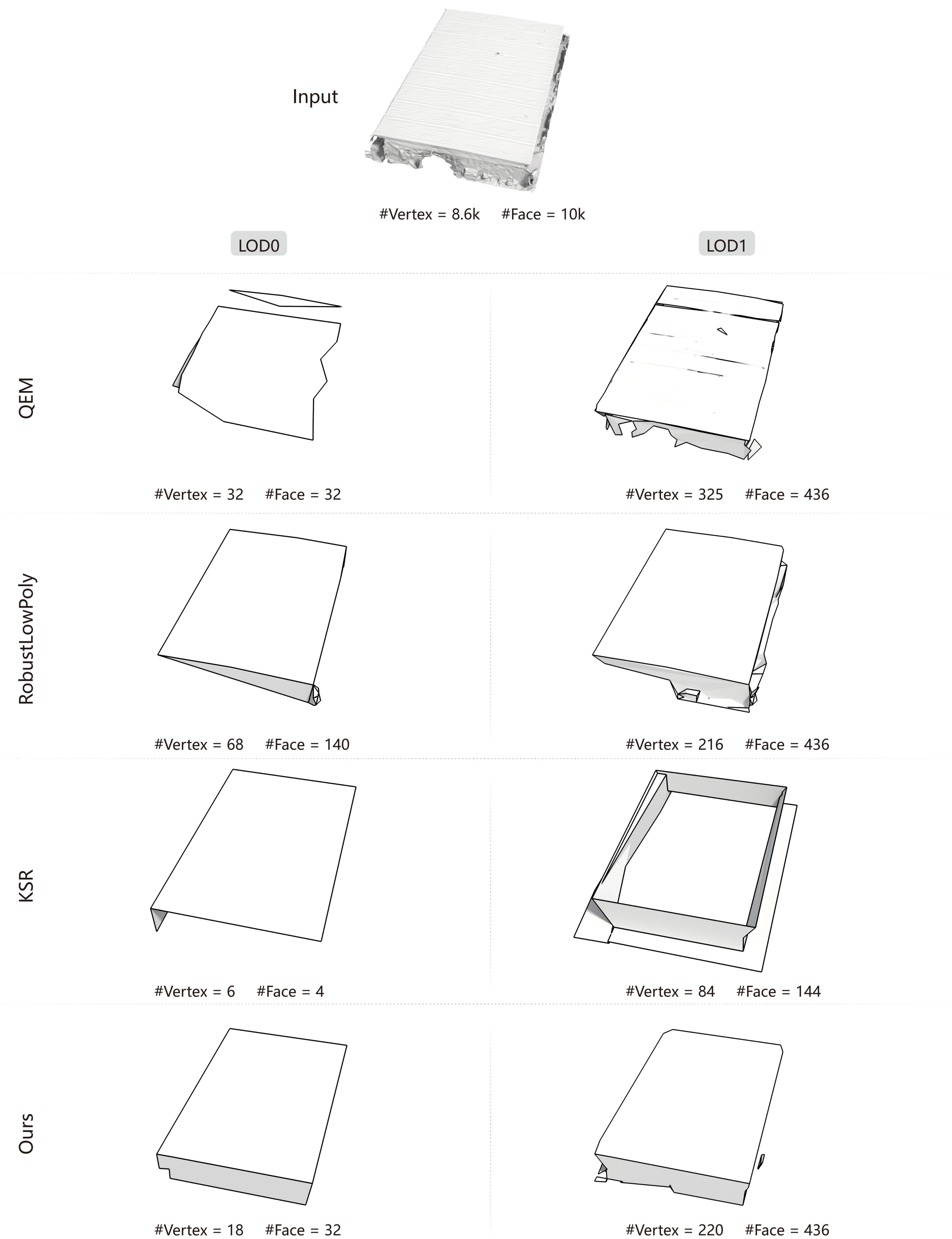

Fig. 21. Qualitative comparisons of randomly selected cases (18/30).

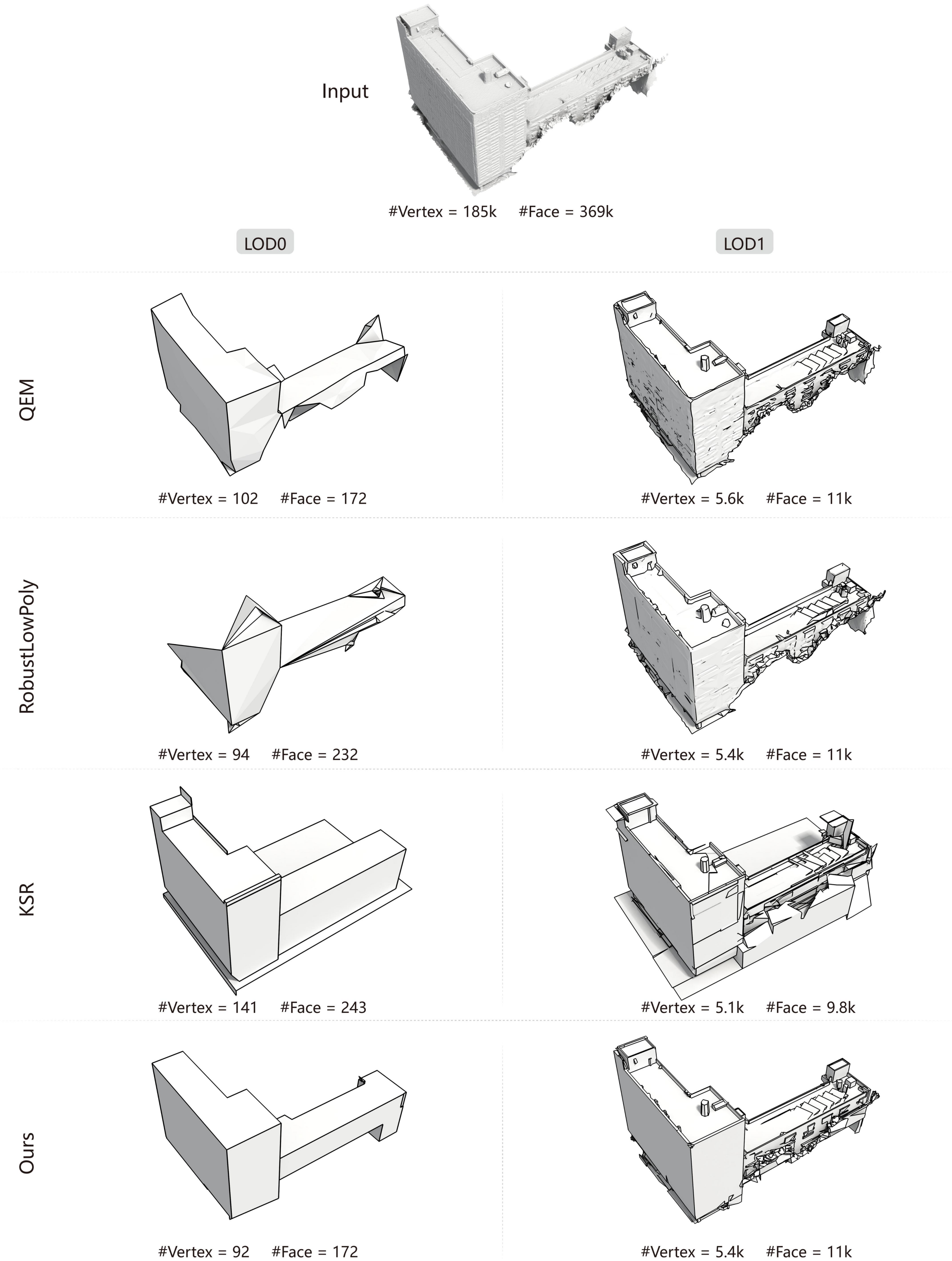

Fig. 22. Qualitative comparisons of randomly selected cases (19/30).

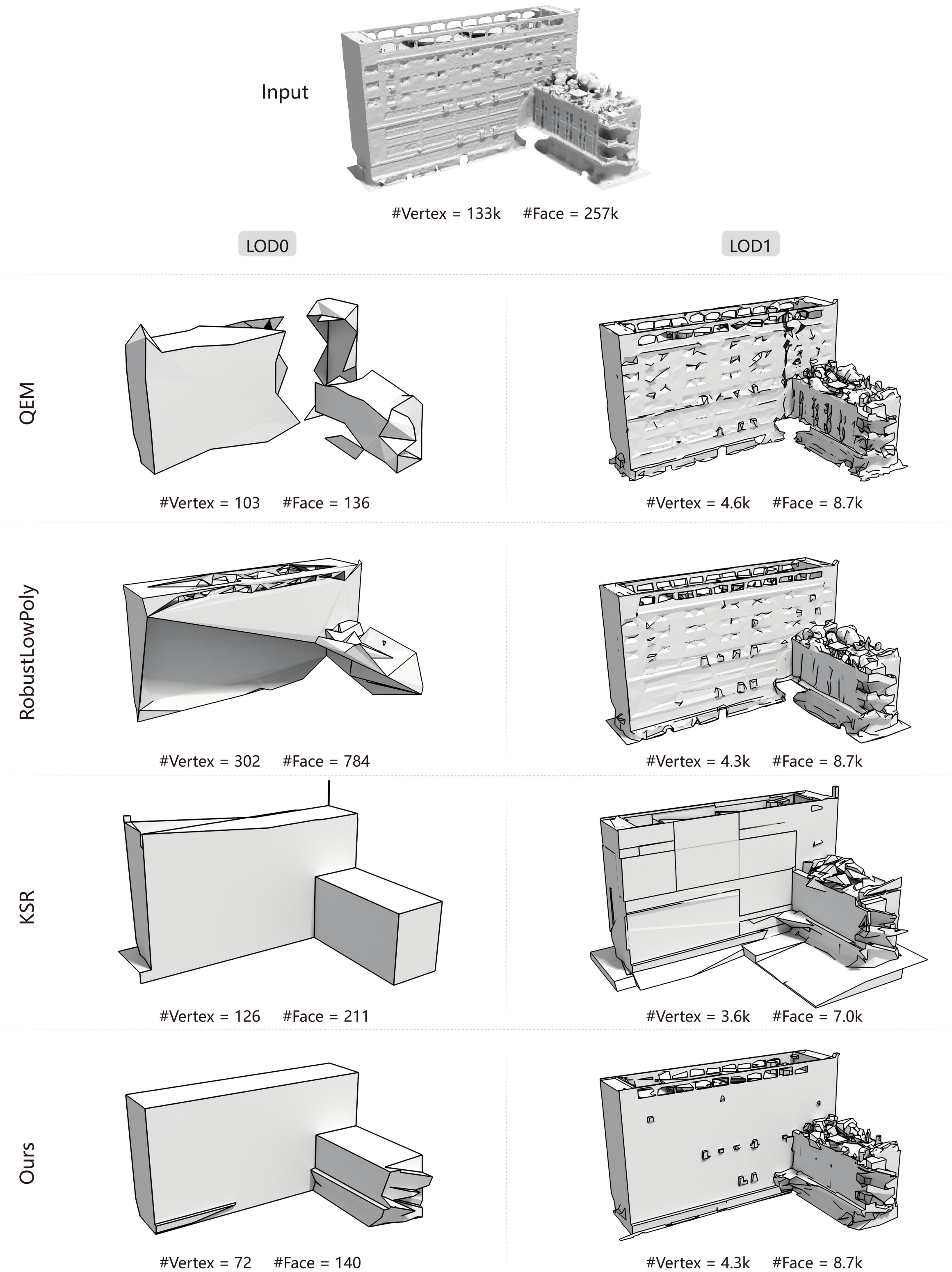

Fig. 23. Qualitative comparisons of randomly selected cases (20/30).

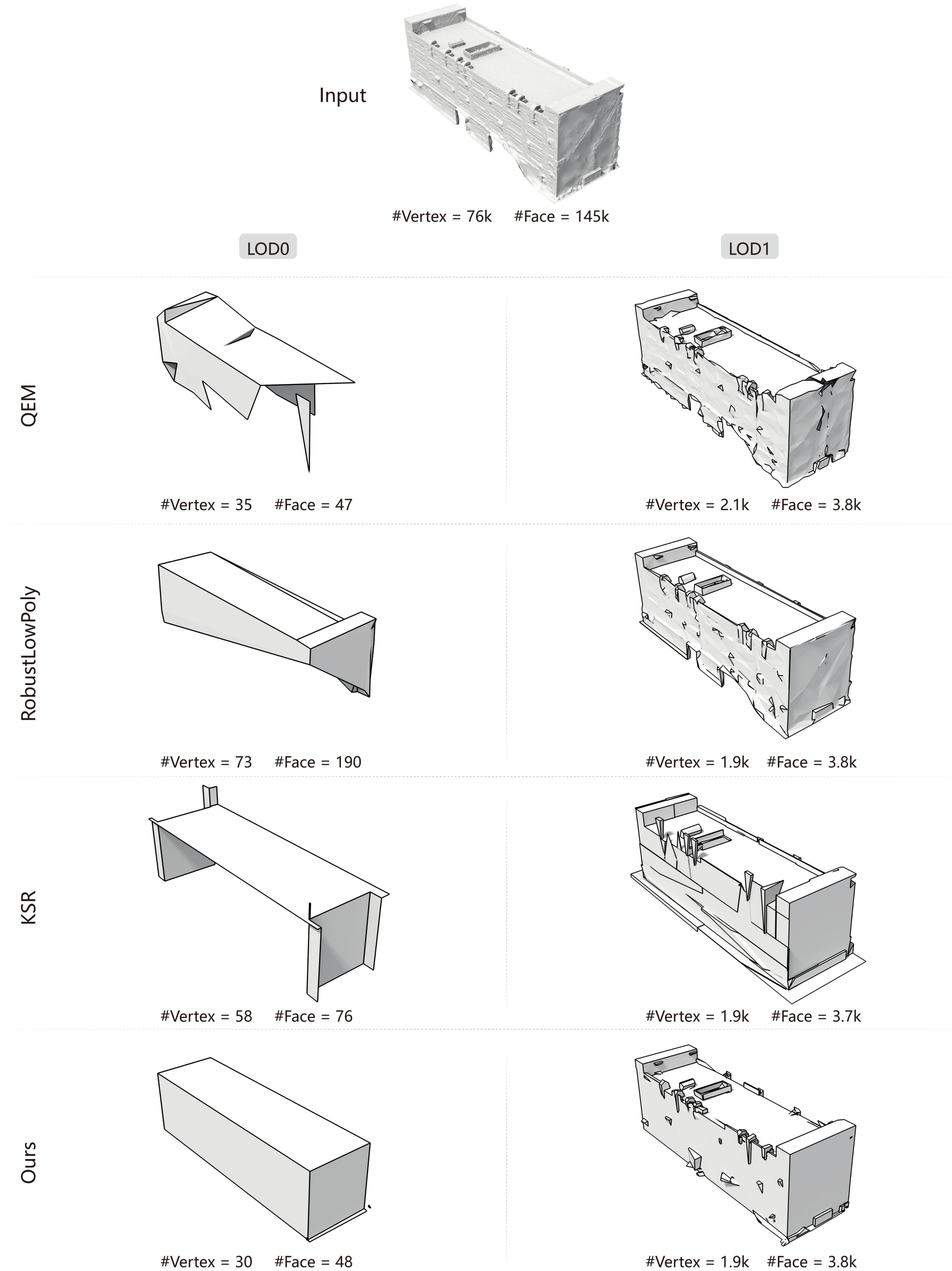

Fig. 24. Qualitative comparisons of randomly selected cases (21/30).

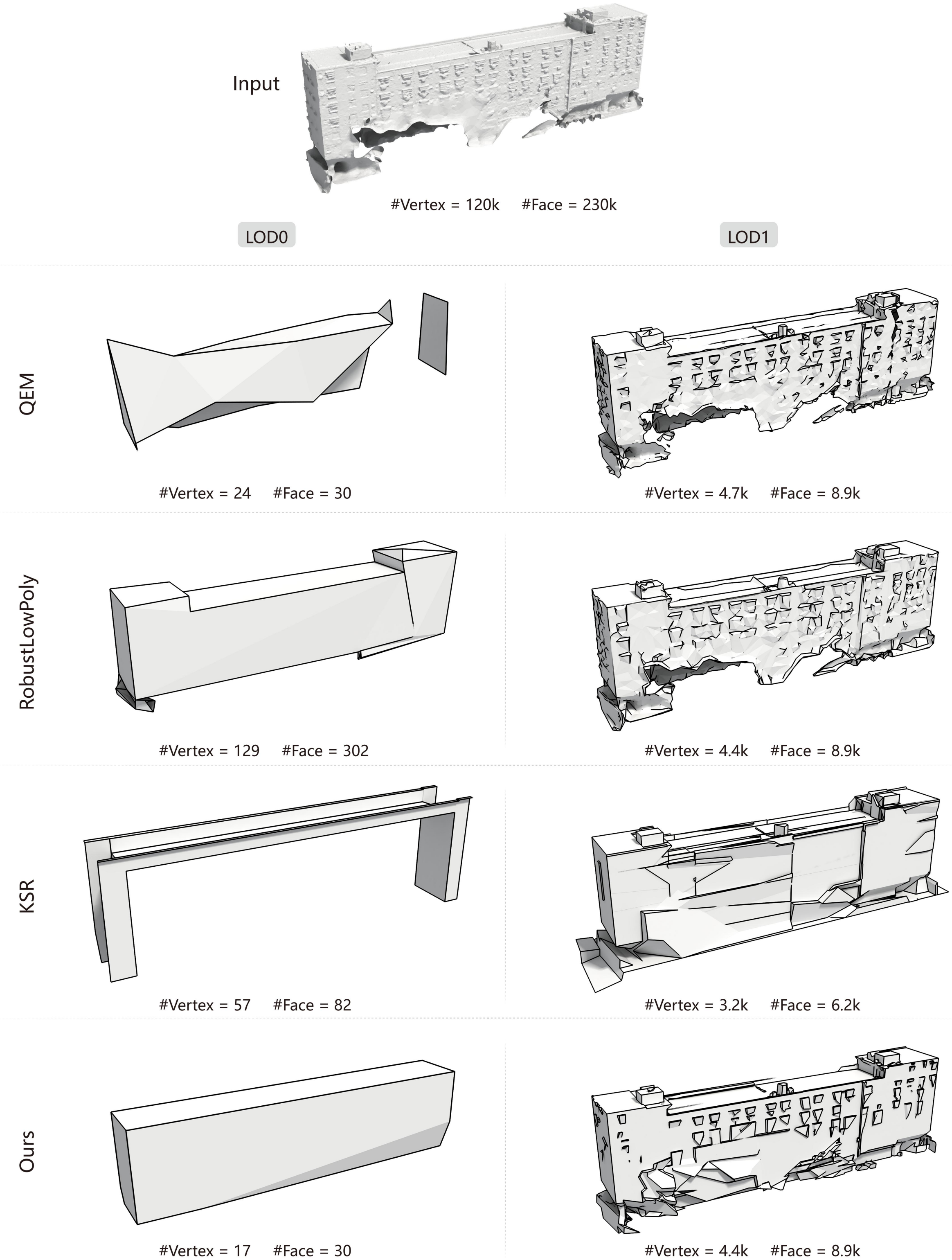

Fig. 25. Qualitative comparisons of randomly selected cases (22/30).

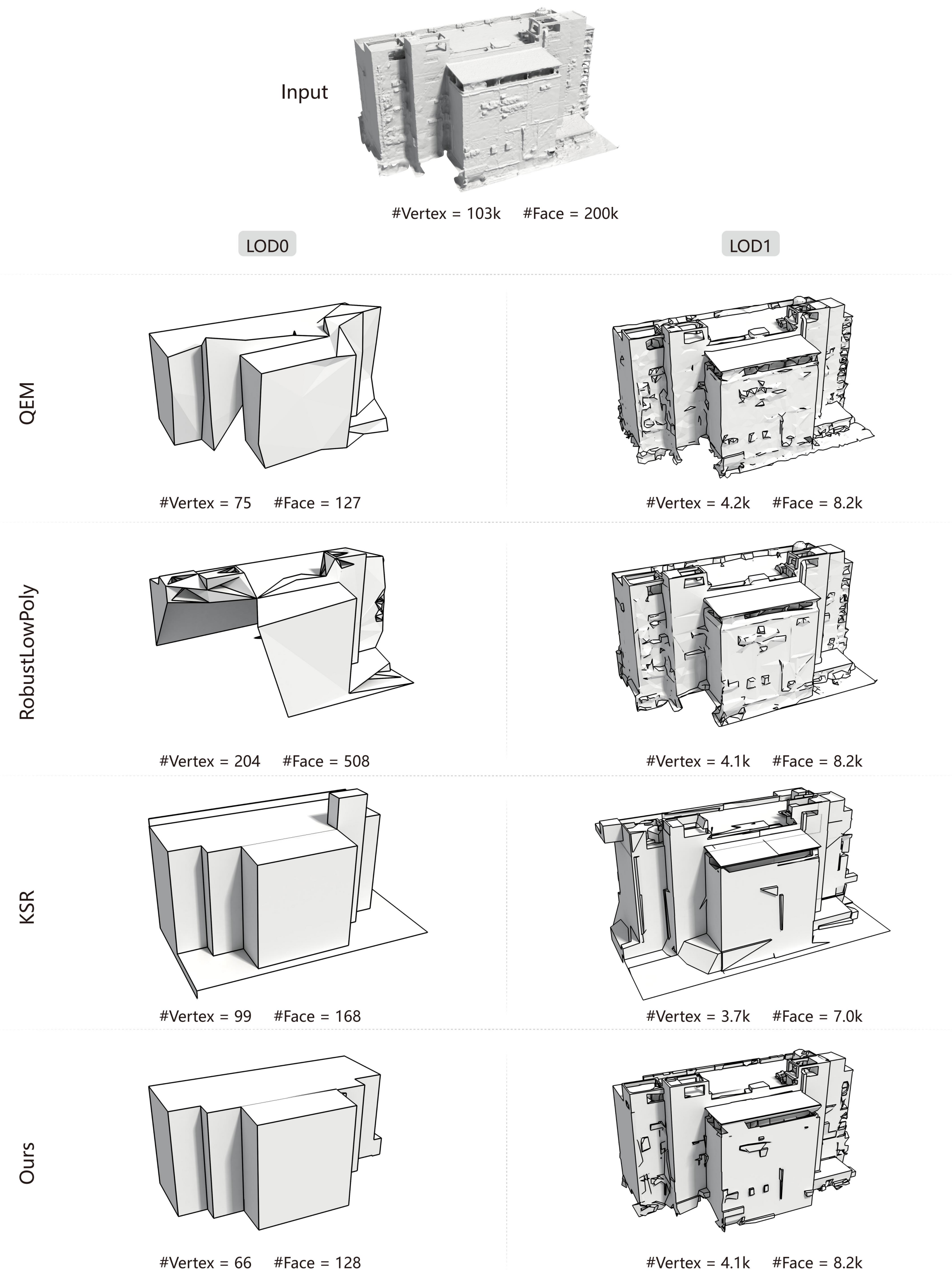

Fig. 26. Qualitative comparisons of randomly selected cases (23/30).

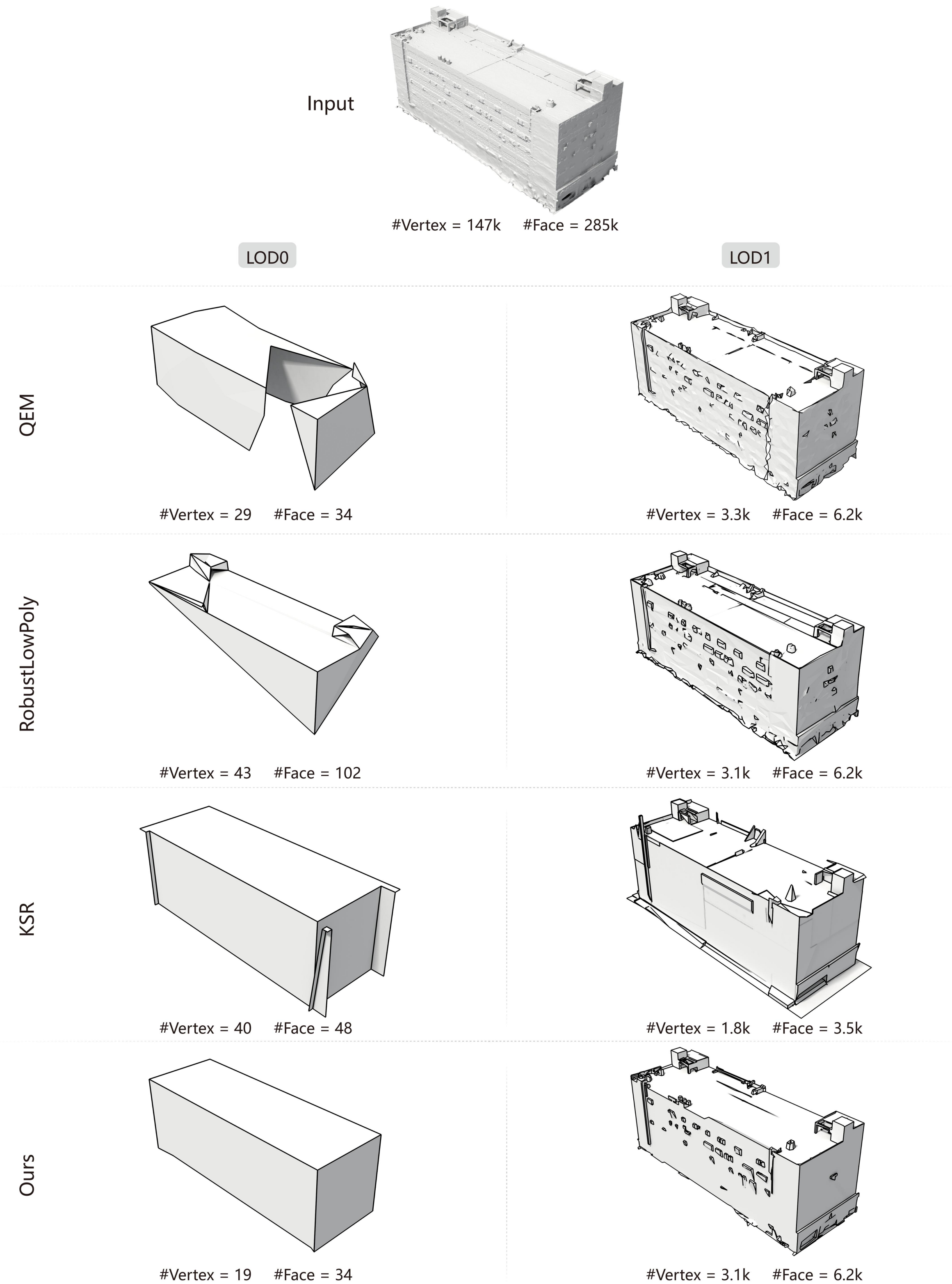

Fig. 27. Qualitative comparisons of randomly selected cases (24/30).

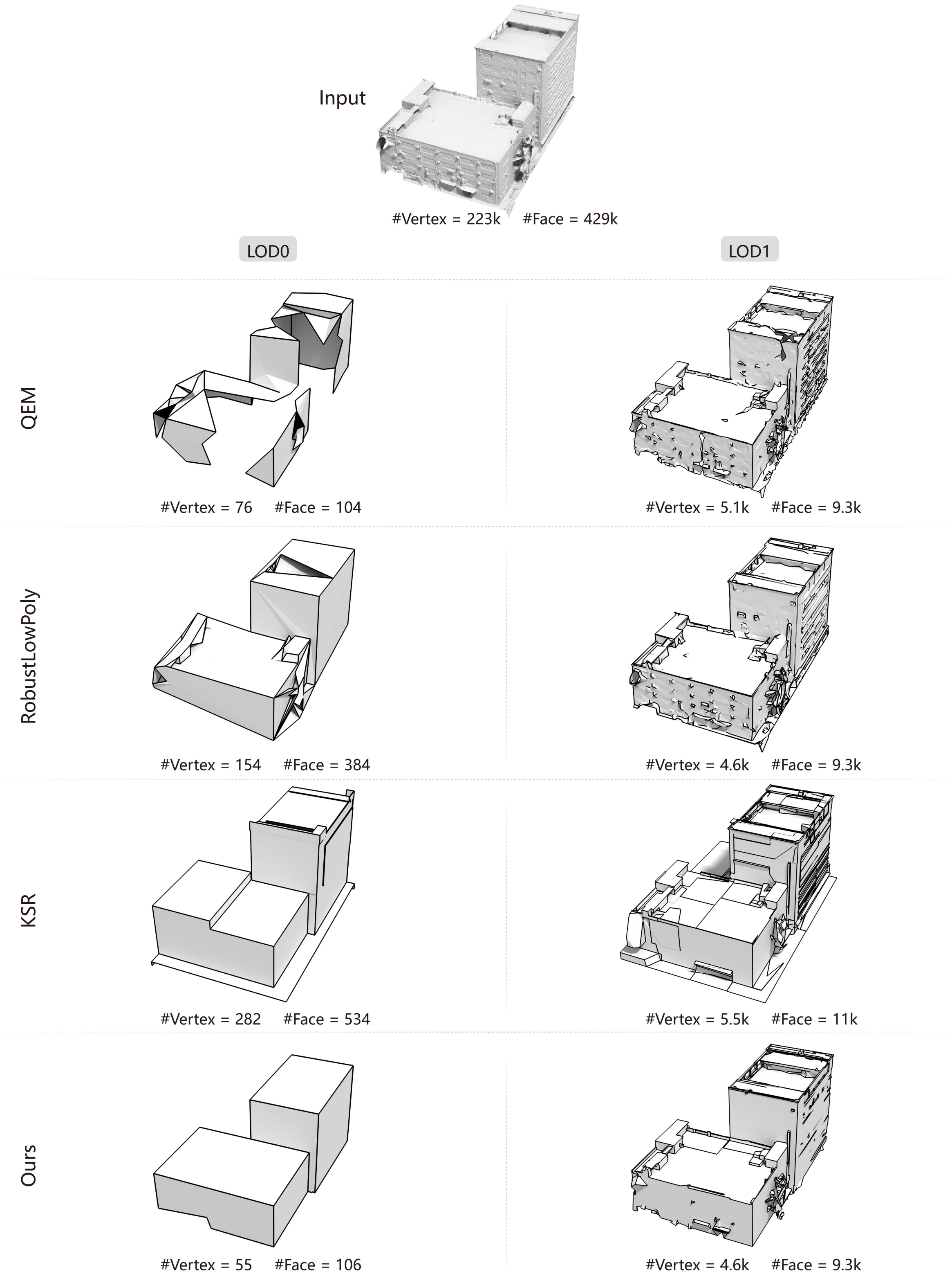

Fig. 28. Qualitative comparisons of randomly selected cases (25/30).

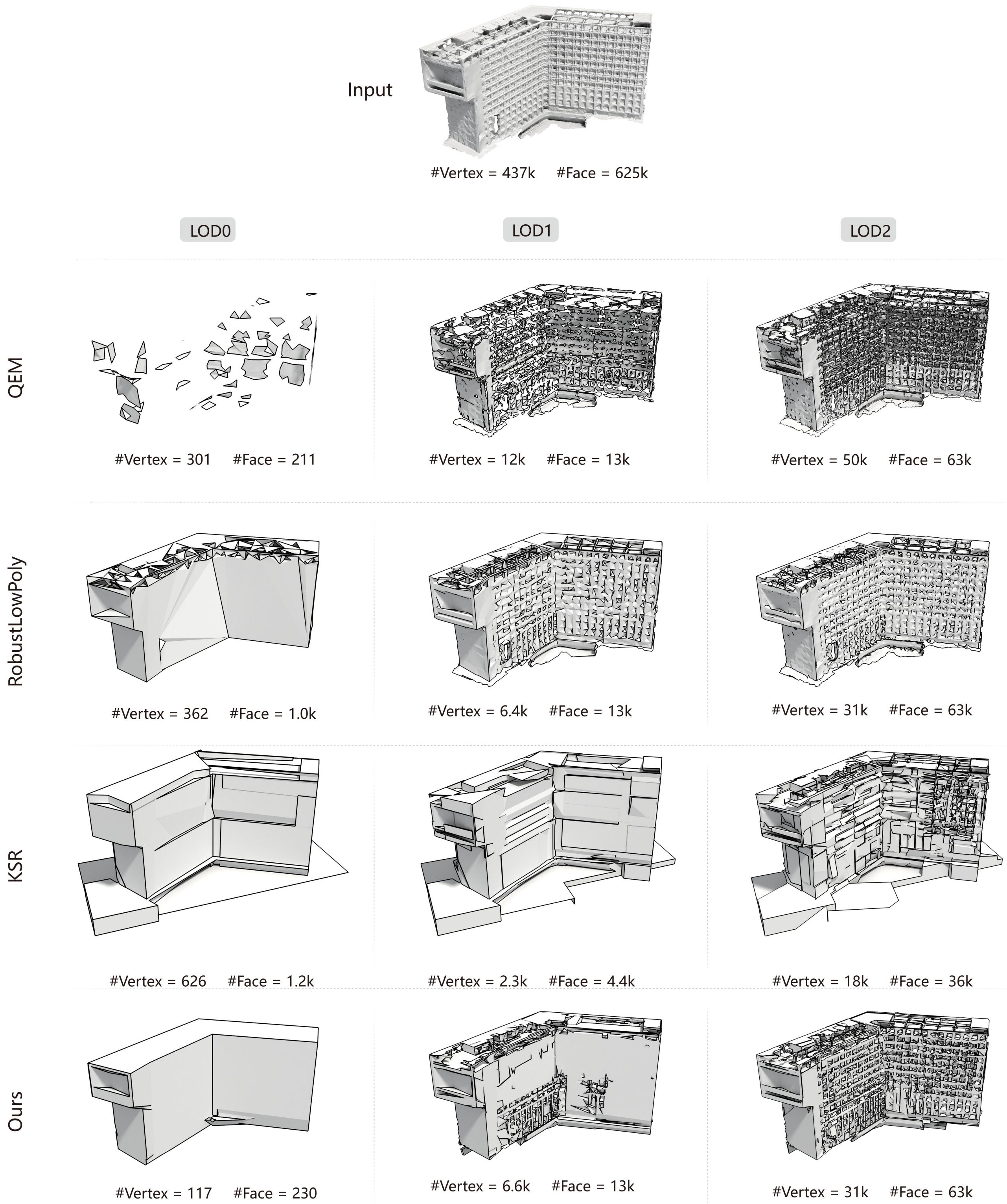

Fig. 29. Qualitative comparisons of randomly selected cases (26/30).

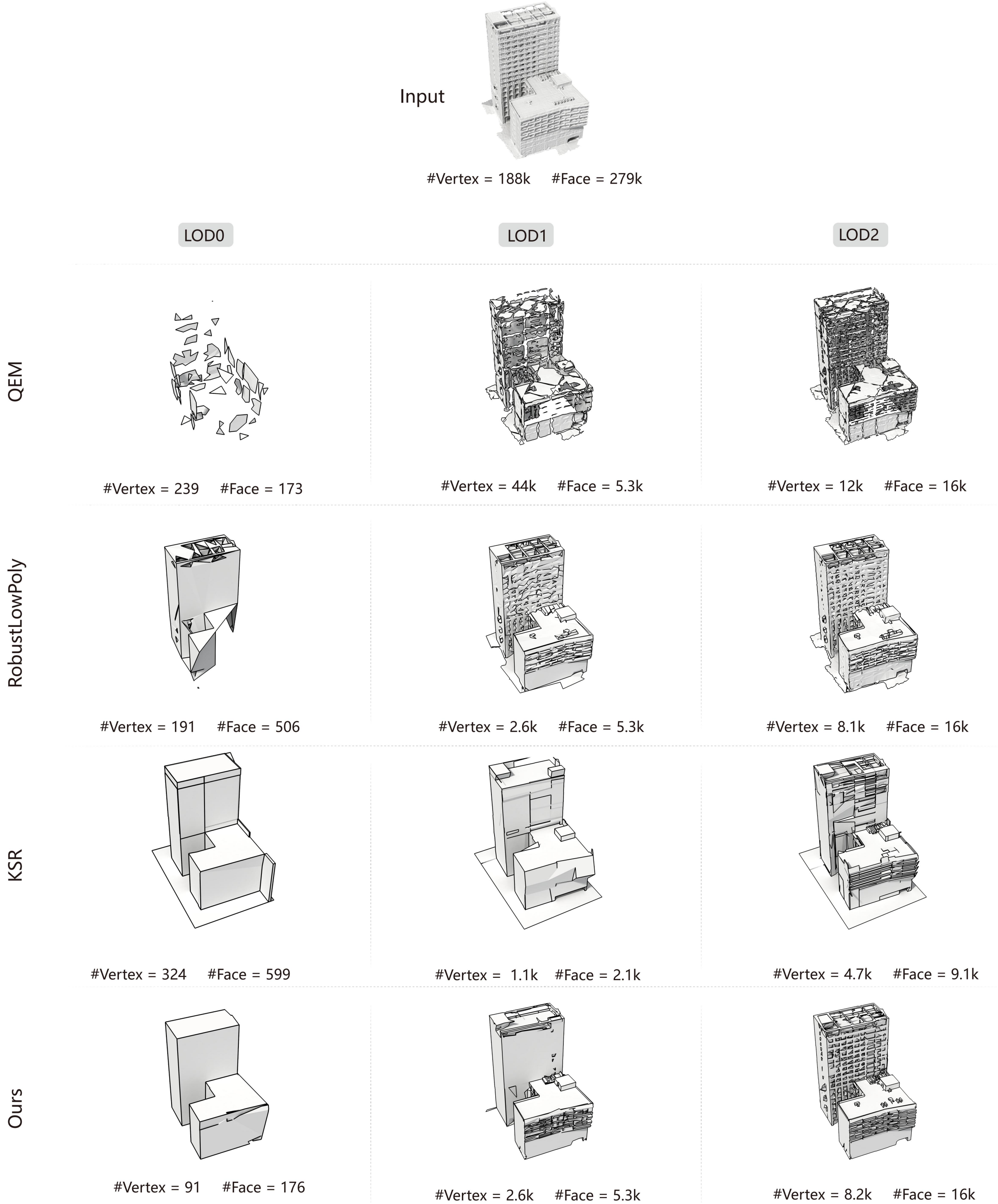

Fig. 30. Qualitative comparisons of randomly selected cases (27/30).

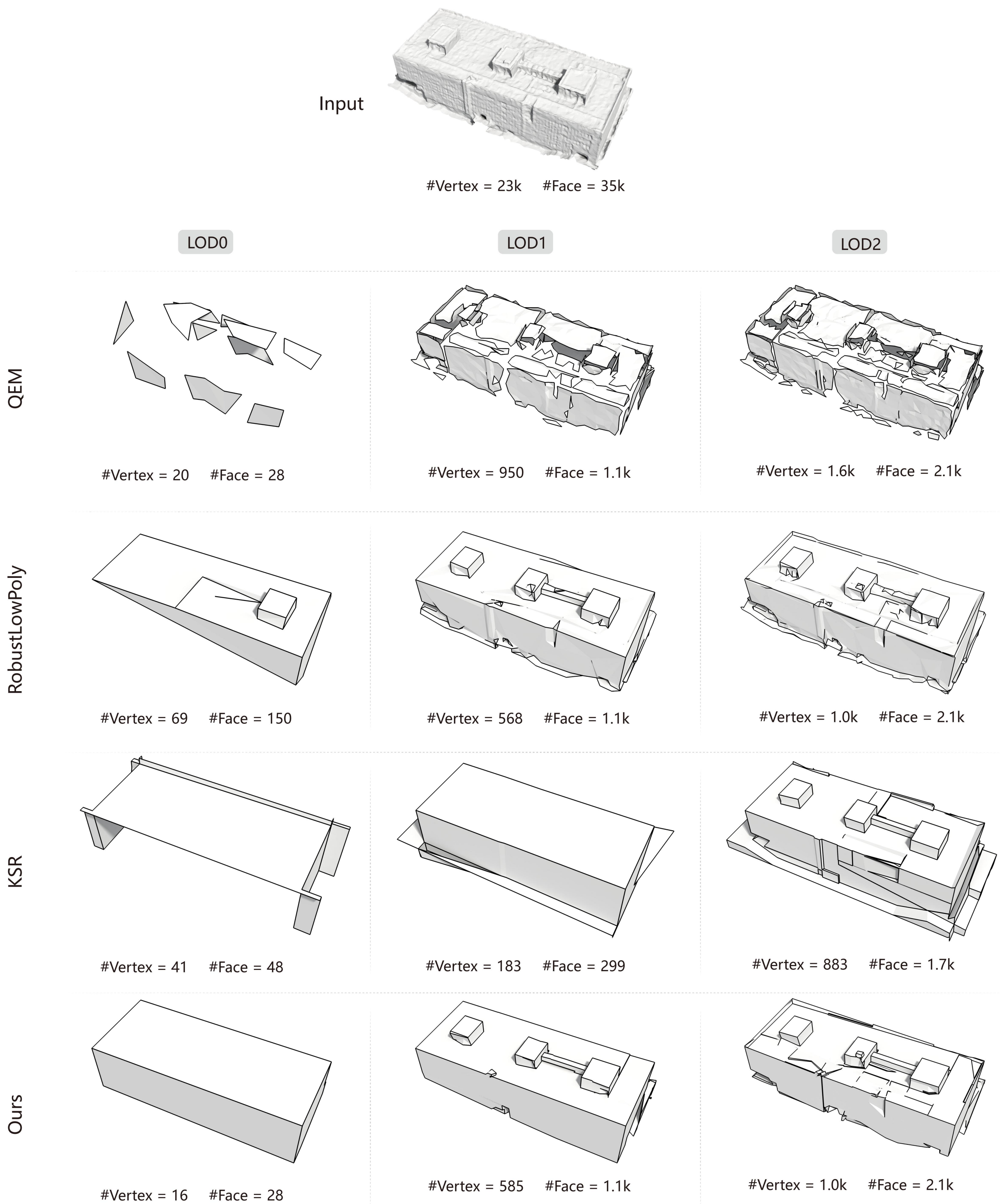

Fig. 31. Qualitative comparisons of randomly selected cases (28/30).

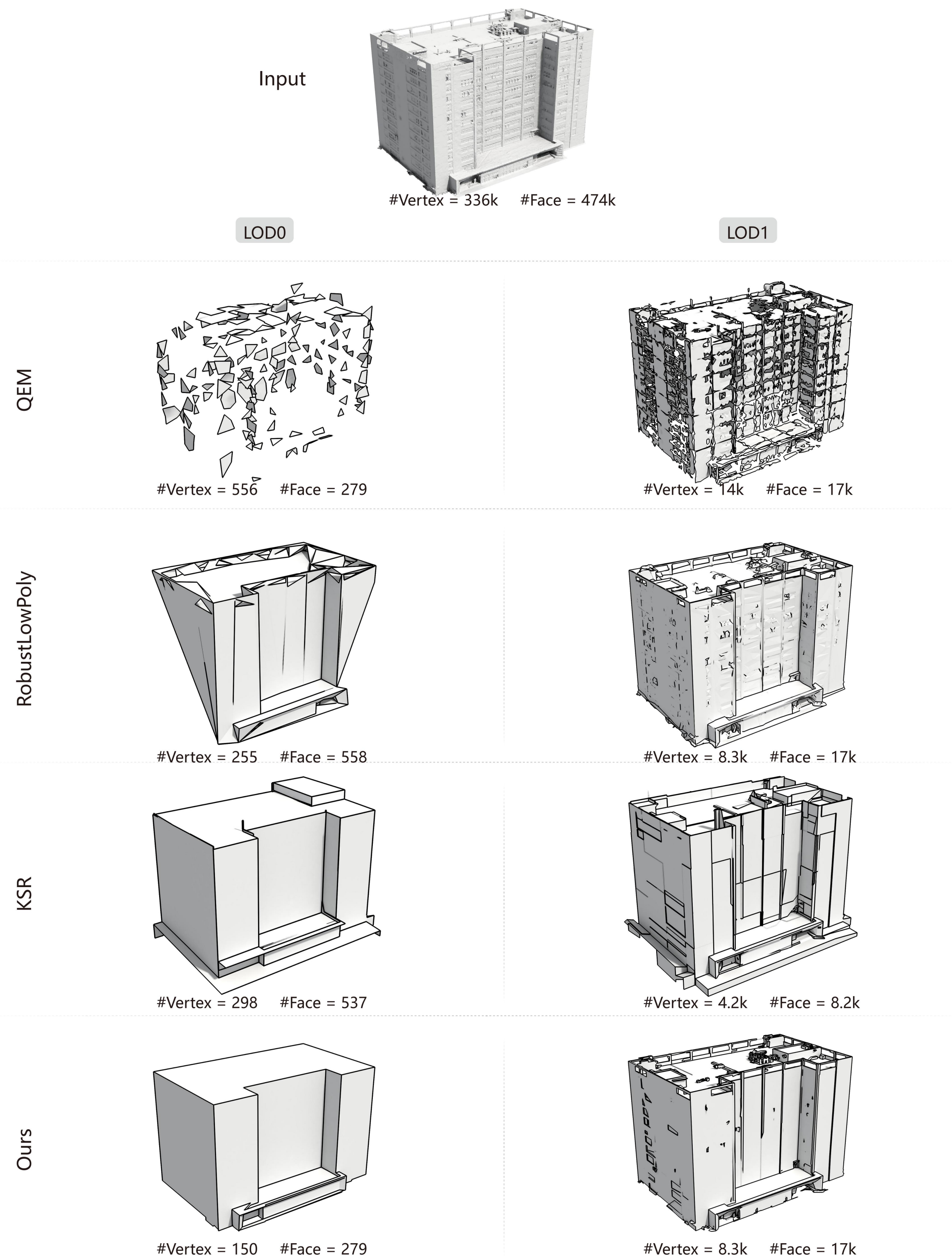

Fig. 32. Qualitative comparisons of randomly selected cases (29/30).

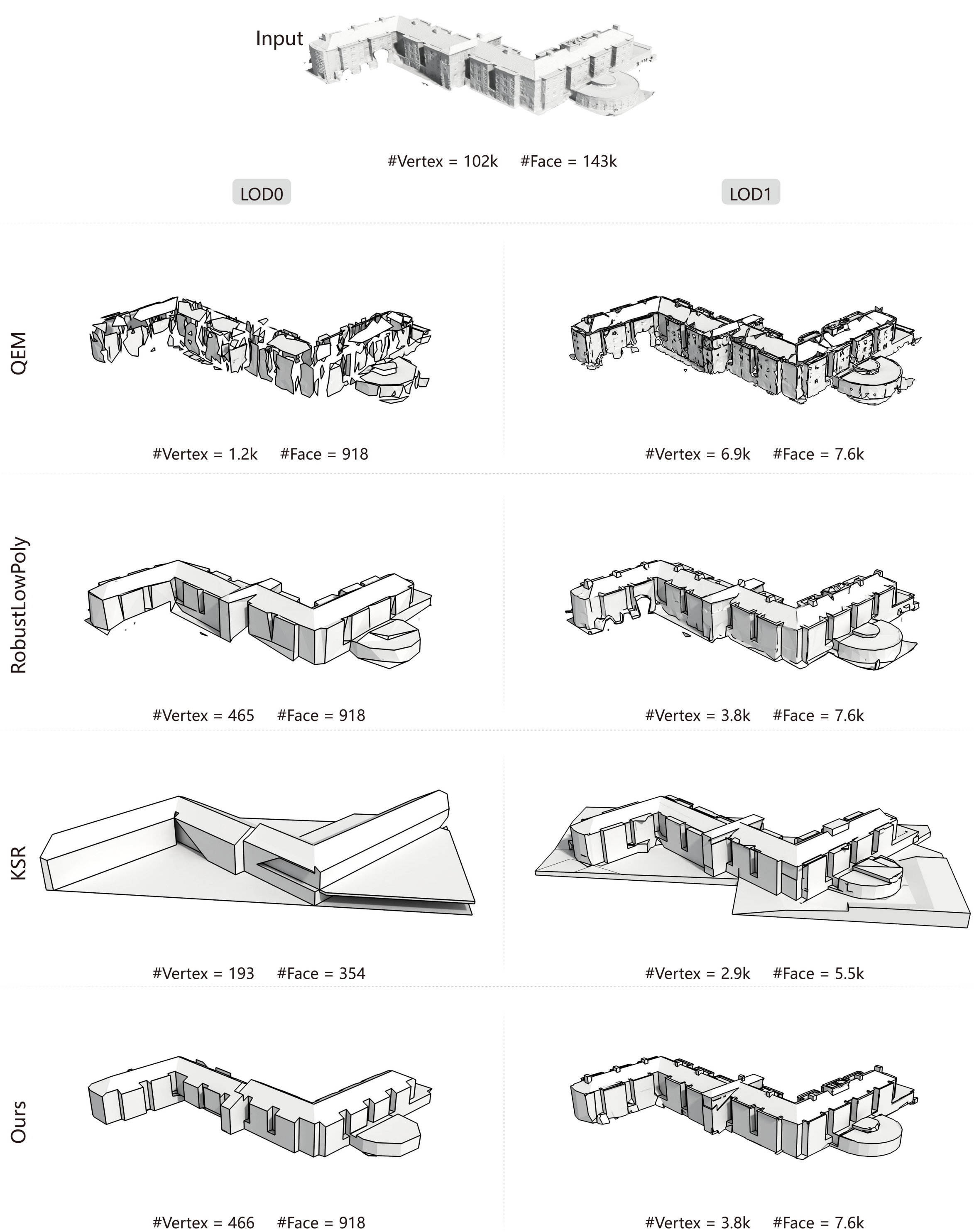

Fig. 33. Qualitative comparisons of randomly selected cases (30/30).

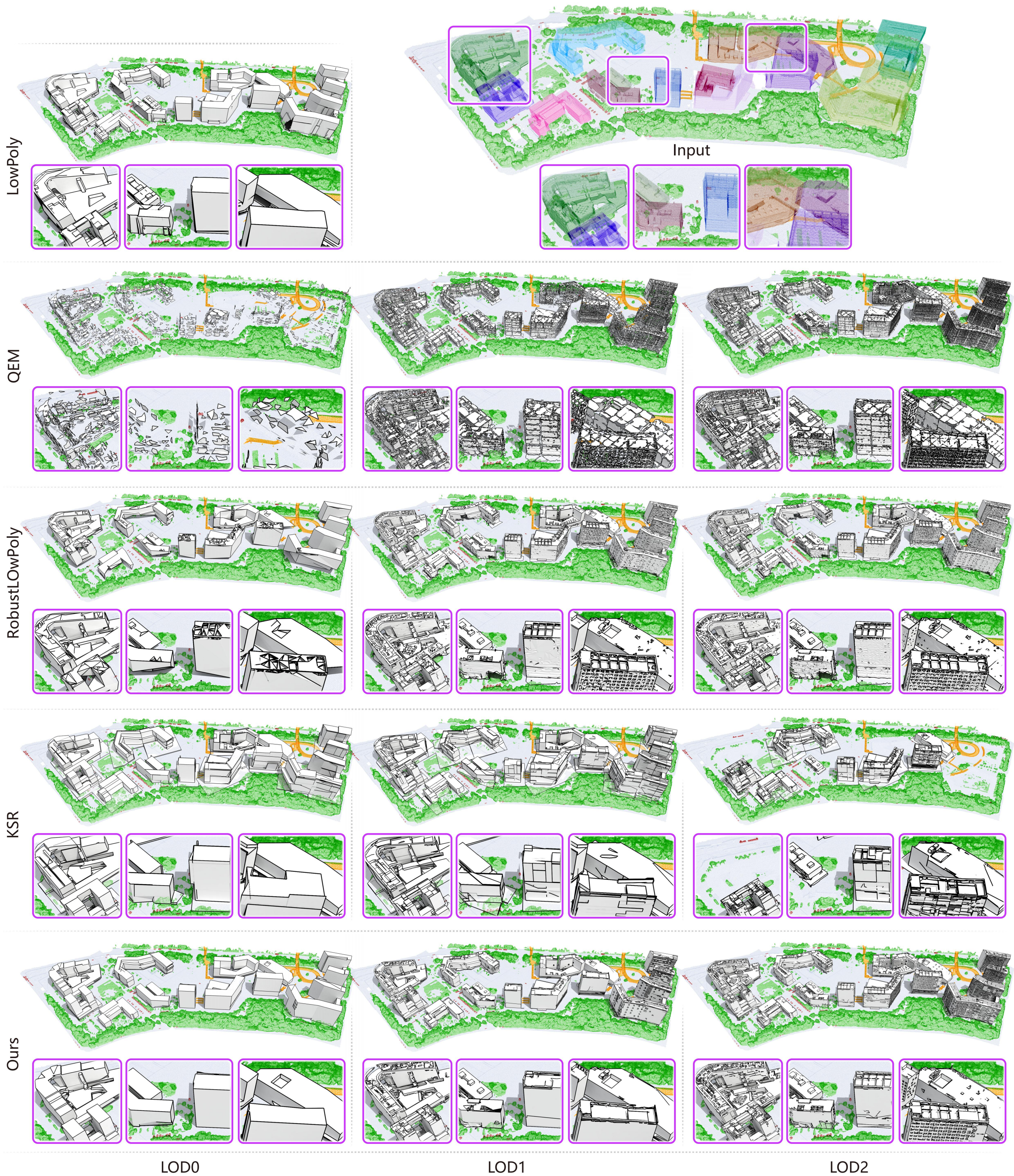

Fig. 34. Qualitative comparisons of *Research Center*.

Fig. 35. Qualitative comparisons of *Town*.

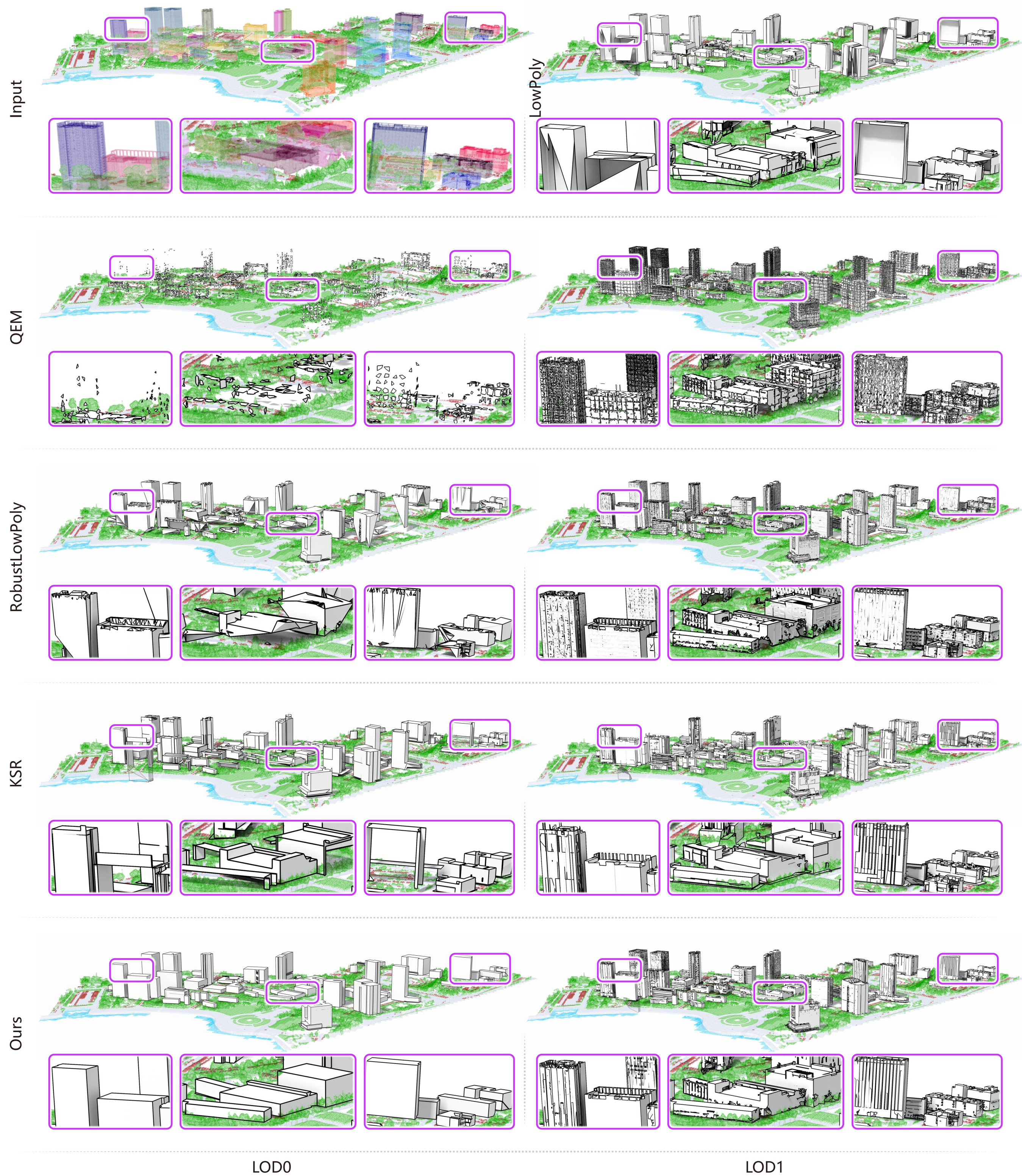

Fig. 36. Qualitative comparisons of *Metropolis*.

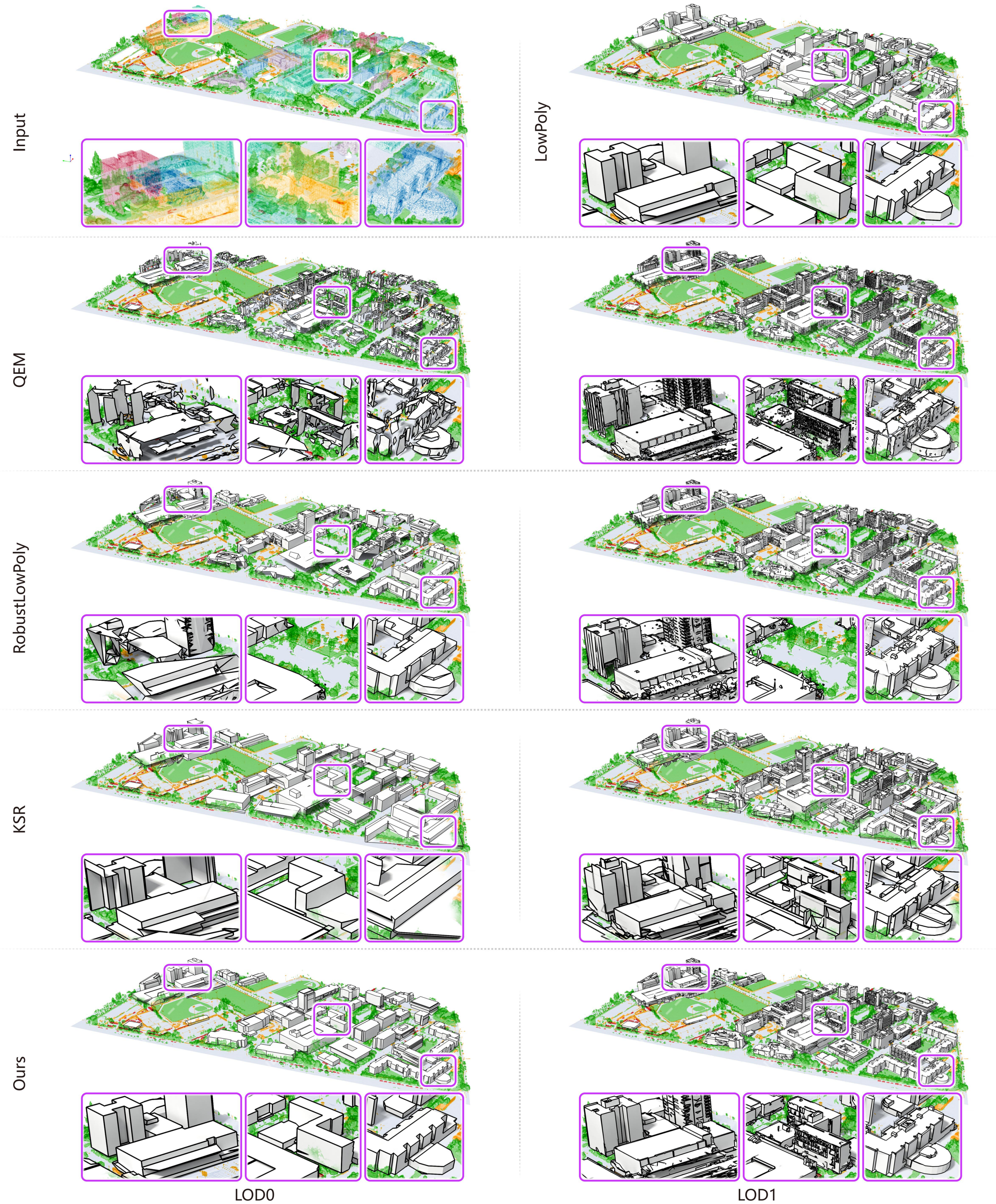

Fig. 37. Qualitative comparisons of *Campus*.

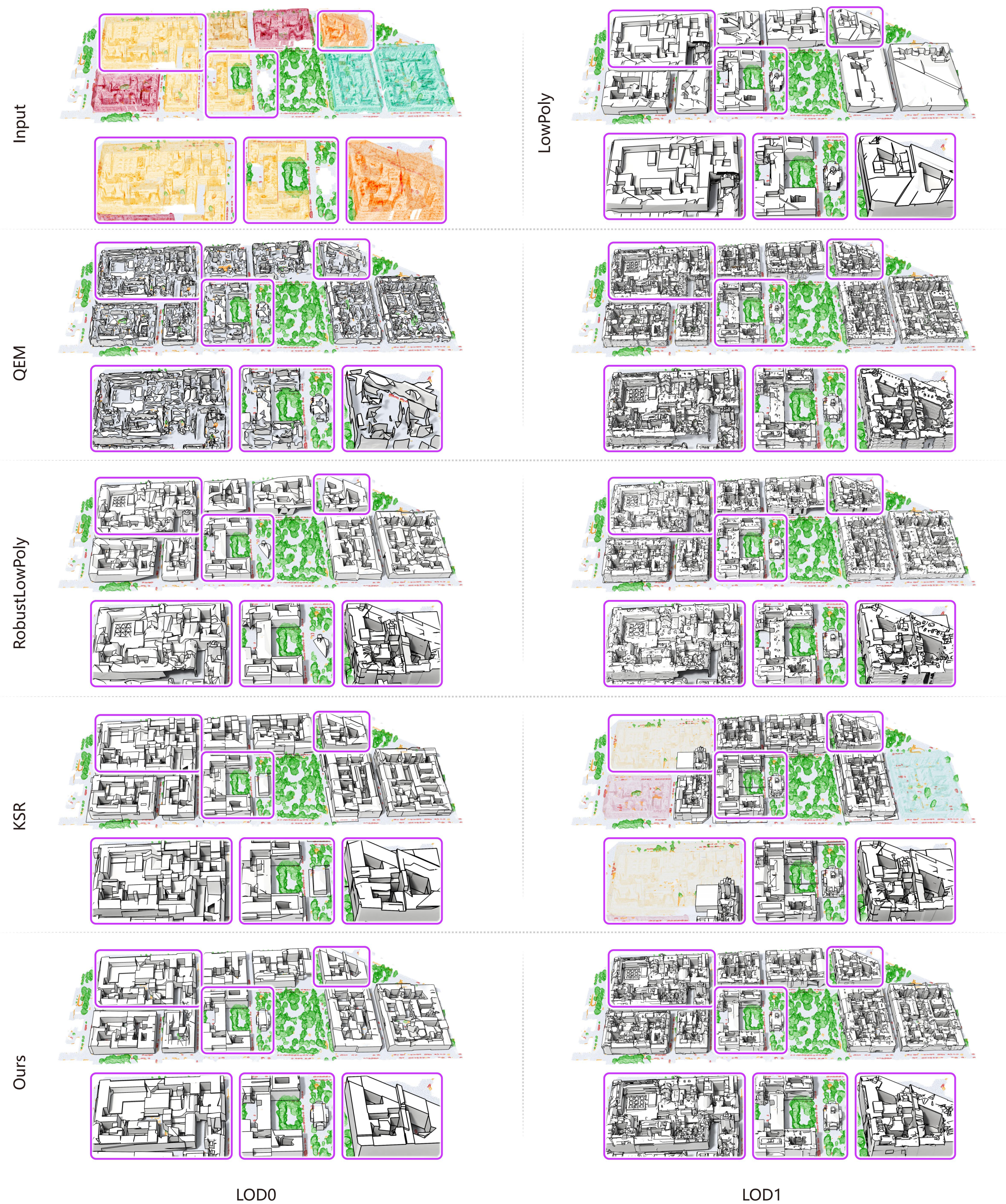

Fig. 38. Qualitative comparisons of *European City*.

Fig. 39. Qualitative comparisons of *Suburbia*.