Supplementary Materials

Jiawei Lu[1,2*], **Yingpeng Zhang**[2*†], **Zengjun Zhao**[2], **He Wang**[3], **Kun Zhou**[1], **Tianjia Shao**[1†]

[1] State Key Lab of CAD&CG, Zhejiang University
[2] Tencent IEG
[3] AI Centre, Computer Science, University College London
lujiawei23@gmail.com, yingpengzhang@gmail.com, zengjunzhao@tencent.com,
he_wang@ucl.ac.uk, kunzhou@zju.edu.cn, tjshao@zju.edu.cn

Implementation Details

Details of surface space color dilation. We first divide the original UV map into sub-UV islands using equal-sized grids. Next, we calculate the connectivity of sub-UV islands and generate an adjacency matrix. Then, we iteratively traverse the invalid pixels. For each invalid pixel, we first pick candidates from textured pixels based on their relative distance in 3D, the cosine similarity of their vertex normal, and the connectivity recorded by the adjacency matrix. We then calculate the color for the invalid pixel by performing a weighted average of these candidates. We iterate this algorithm until all invalid pixels are filled or reach the max step. The detailed algorithm on surface space color dilation is shown in Algorithm. 1. An illustration of this process is shown in Fig. 3, with an ablation on the surface space color dilation shown in Fig. 2.

More details. We implement our algorithm using an open-source framework: ComfyUI(ComfyUI 2024), and we adopt nvdiffrast (Laine et al. 2020) for rendering. We set the strength of ControlNet as 1.0 in all our experiments.

ALGORITHM 1: UV dilation in surface space

Input:
input UV map U
uv-space spatial position map X
uv-space normal map N
uv-space face index map F
uv-space visibility map M
Parameters: grid size s, dilation distance threshold d_{th}, dilation angle threshold a_{th}, iterations $iter$, number of nearest neighbors n
Output: UV map after dilation U

$I_{ori} \leftarrow get\_original\_uv\_island(F)$
$I_{grid} \leftarrow get\_grid\_uv\_island(F, s)$
$M_{adj} \leftarrow get\_adjacency\_matrix(F, I_{grid}, I_{ori})$
$P, Q \leftarrow get\_valid\_invalid\_points(M)$
for $i = 1, 2, \ldots, iter$ **do**
  for *each* $q \in Q$ **do**
    $A = I_{grid}[q]$
    $q_n \leftarrow KNN(q, P, n)$
    for *each* $q_k \in q_n$ **do**
      $B = I_{grid}[q_k]$
      $dist = ||X[q] - X[q_k]||_2$
      $angle = angle\_between(N[q], N[q_k])$
      if $q_k \notin Q$ ***and*** $angle < a_{th}$ ***and*** $M_{adj}[A][B] == True$ ***and*** $dist < d_{th}$ **then**
        $w_k = 1 - (dist/d_{th})^2$
      else
        $w_k = 0$
      end
    end
    $w = \sum_{q_k \in q_n} w_k$
    if $w \neq 0$ **then**
      $U[q] = \frac{1}{w} \sum_{q_k \in q_n} (U[q_k] * w_k)$
      remove q from Q
    end
  end
end

More Results

We show extra ablation experiments on local attention in Fig. 1. This example shows the ablation results on different attention mechanisms in multi-view generation without latent merge. Our local attention achieves the best multi-view consistency while preserving rich details close to the original unconstrained diffusion image(row 1). In addition, we provide more comparison results with different methods in Fig. 4, 5, 6. More results by our methods on various meshes and styles can be viewed in Fig. 7, 8, 10, 11, 12. We highly recommend readers to watch the videos for more results.

References

ComfyUI. 2024. ComfyUI. https://github.com/comfyanonymous/ComfyUI/.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2023. Objaverse: A universe of annotated

*These authors contributed equally.
†Corresponding author.

Figure 1: Ablation results on different attention mechanisms in multi-view generation. Each view attends to its neighbors (top), each view attends to all other views (middle), our **local attention** (bottom) achieves the best multi-view consistency while preserving rich details.

3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13142–13153.

Laine, S.; Hellsten, J.; Karras, T.; Seol, Y.; Lehtinen, J.; and Aila, T. 2020. Modular Primitives for High-Performance Differentiable Rendering. *ACM Transactions on Graphics*, 39(6).

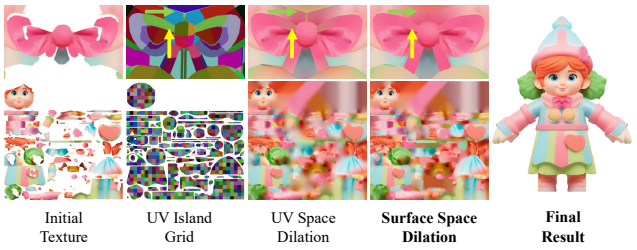

Figure 2: An illustration of surface space color propagation algorithm for texture completion. Our method propagates valid texture color in surface space instead of UV space. This effectively addresses inaccurate color propagation when two points are proximate to each other in 3D but situated on remote UV islands (green arrow), or located on nearby UV islands but having a large 3D distance (yellow arrow).

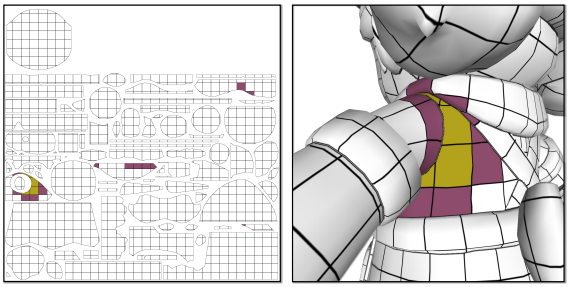

Figure 3: An illustration on our texture dilation algorithm. The yellow area can be influenced by the neighbor regions in surface space. Note how the colors can be propagate between distant UV islands.

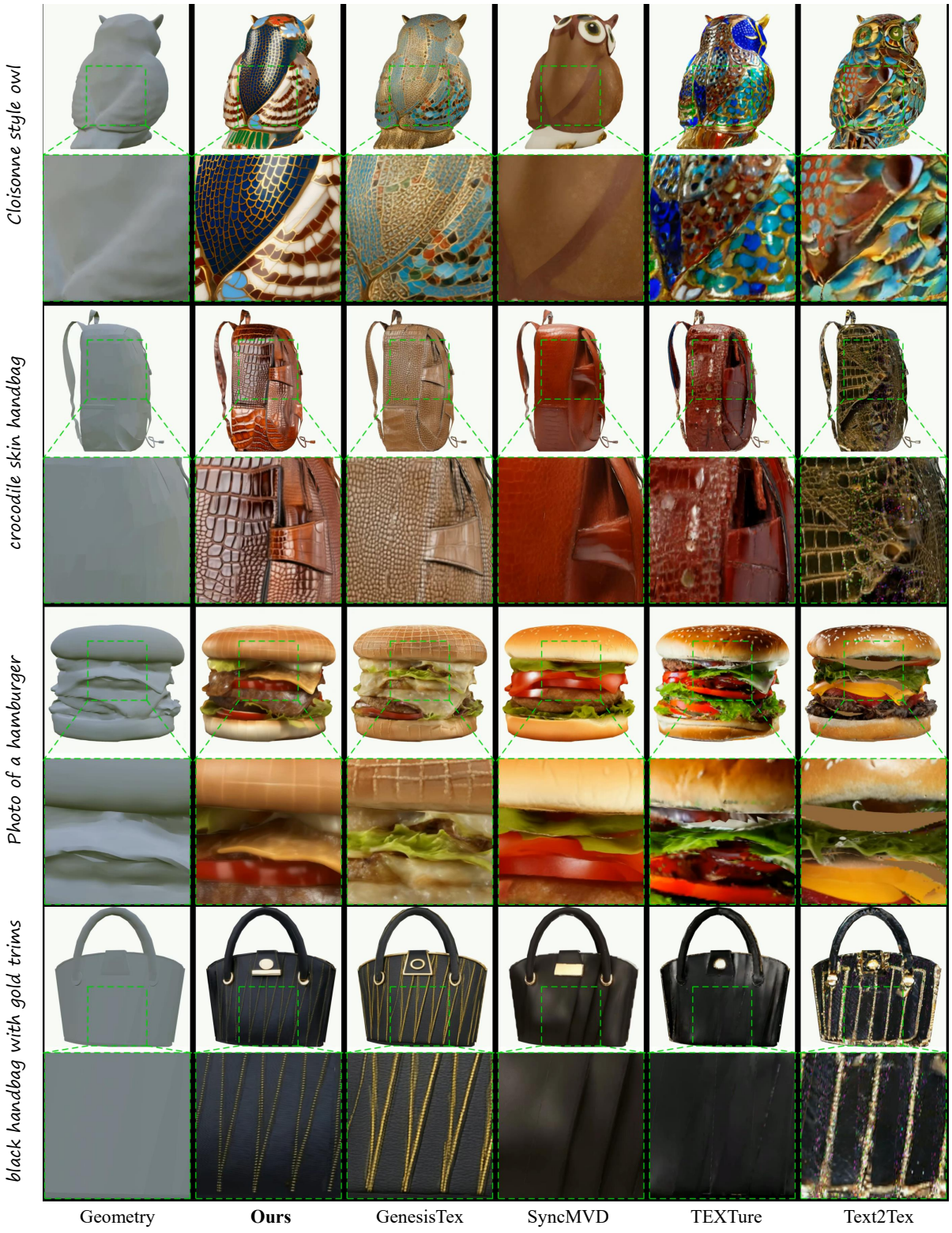

Figure 4: More comparison results with different methods.

Figure 5: More comparison results with different methods.

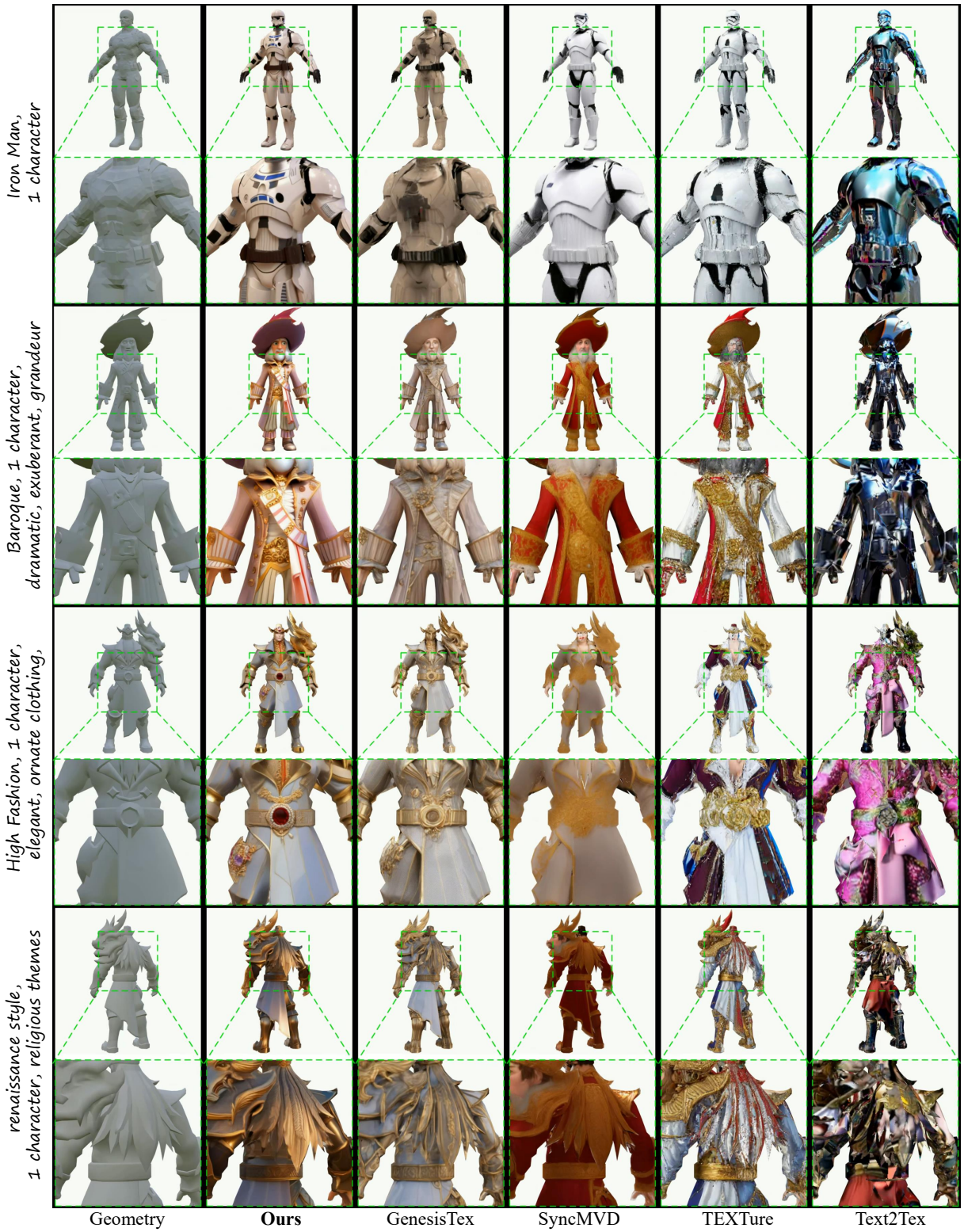

Figure 6: More comparison results with different methods.

Figure 7: More results on meshes from objaverse(Deitke et al. 2023).

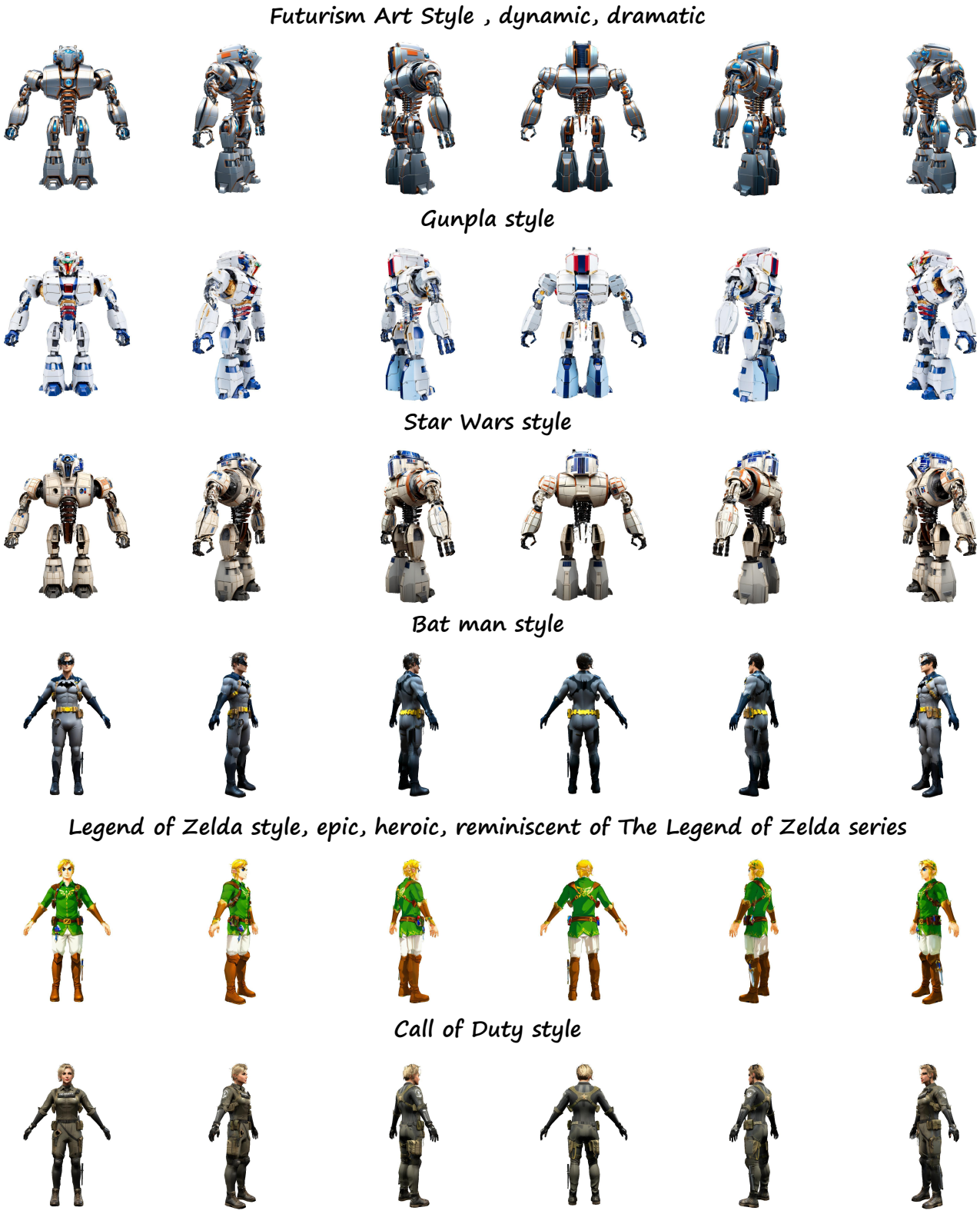

Figure 8: More results on meshes from objaverse(Deitke et al. 2023).

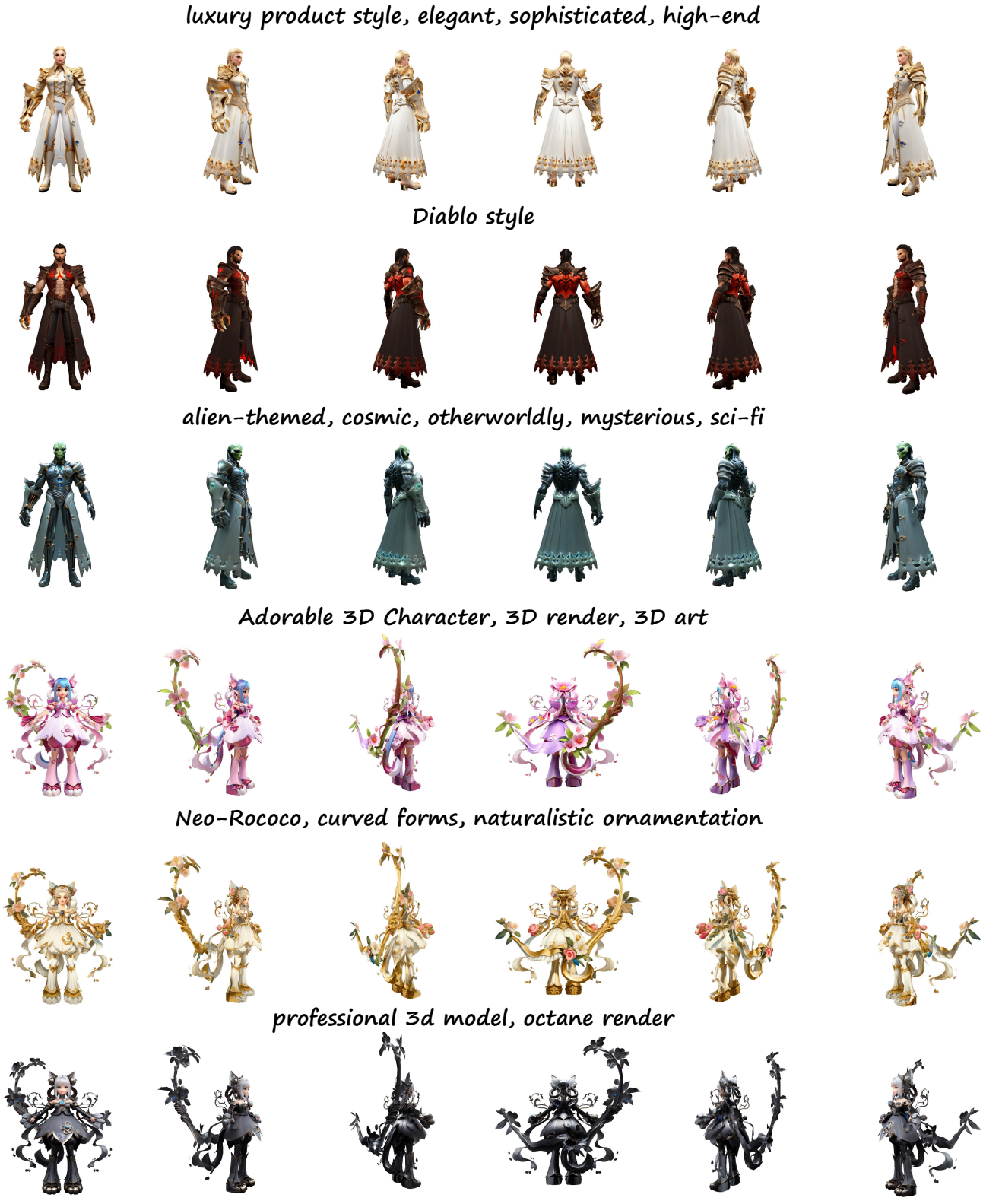

Figure 9: More results on meshes from objaverse(Deitke et al. 2023).

Figure 10: More results on meshes from industrial games.

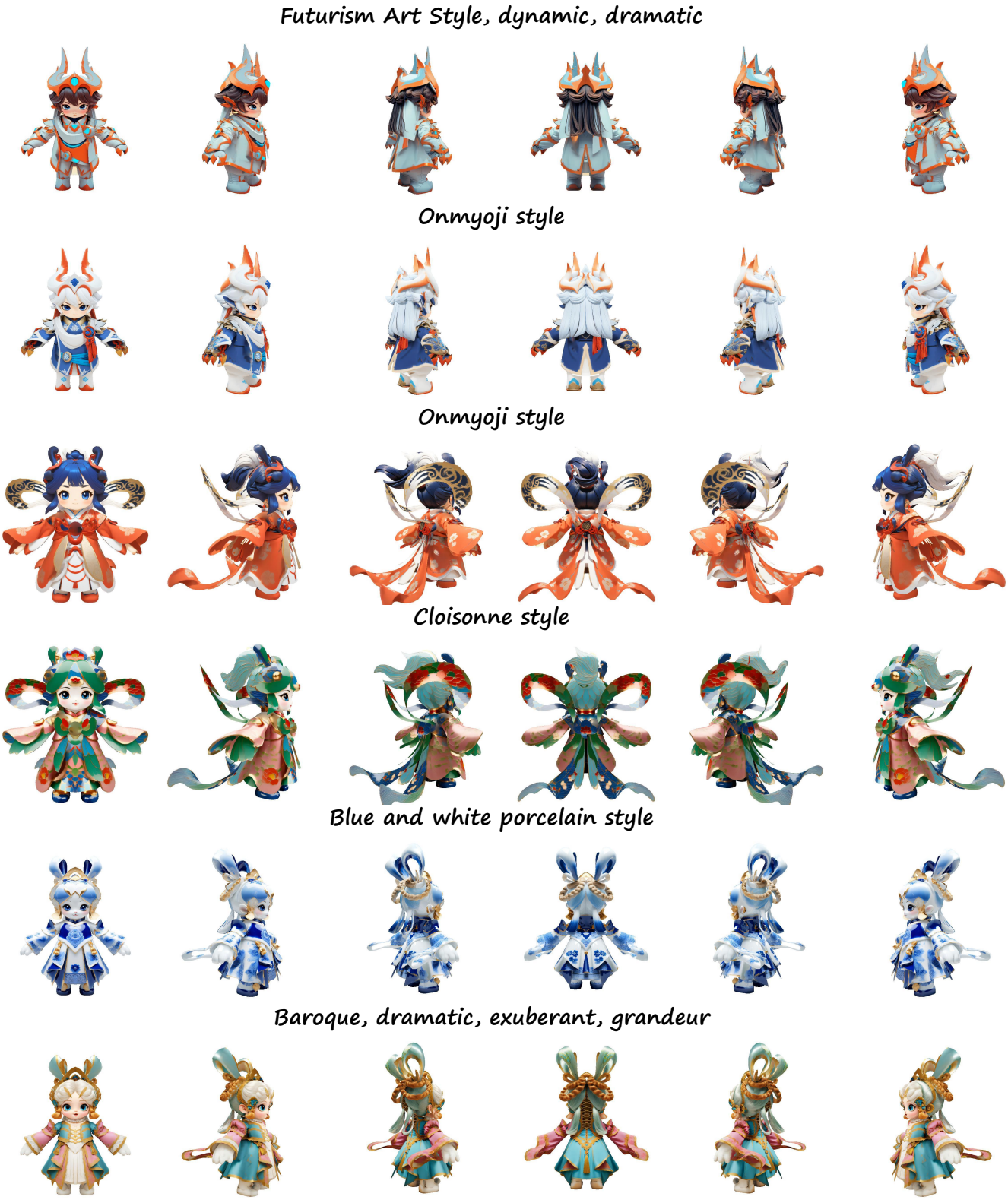

Figure 11: More results on meshes from industrial games.

Figure 12: More results on meshes from industrial games.