Layered Image Vectorization via Semantic Simplification

Supplementary Material

1. Additional Ablations

Optimization of Structure-wise Vectors We investigated the effectiveness of structure-wise vector optimization compared to direct initialization of structure-wise vectors from segmented masks, without optimization. Figure 1 shows with the optimization, the vector boundaries become neater and more refined.

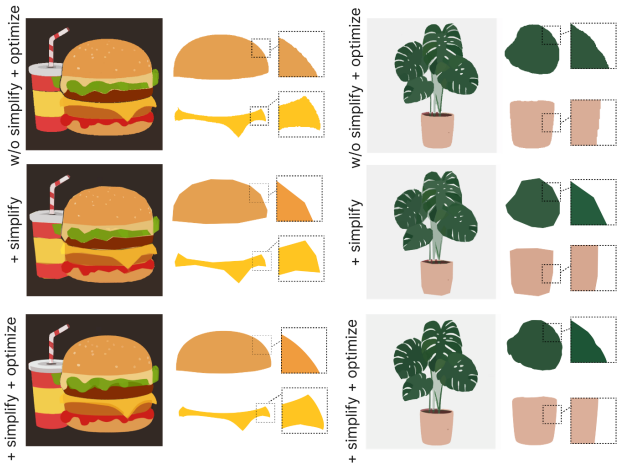

Figure 1. Comparison of structure-wise vectors with and without structure-wise vector optimization.

Overlap Loss We examined the impact of the overlap loss $\mathcal{L}_{overlap}$ in structural construction. Figure 2 shows with overlap loss, the vector boundaries are aligned better than those without the overlap loss.

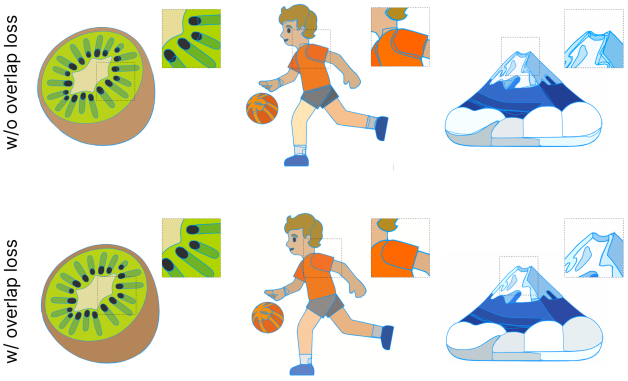

Figure 2. Comparison of structure-wise vectors with and without the *overlap loss*.

2. Implementation Details

Initialization of Vector Primitives Both structure-wise and visual-write vectors are initialized as closed shape of cubic Bézier curves. For initialization, the boundary of the mask (for structure-wise vectors) or the boundary of the top-K connection area (for visual-wise vectors) is simplified using the Douglas–Peucker algorithm. This algorithm reduces the number of points while ensuring the simplified boundary remains within a distance ϵ (ϵ = 5.0 in our work) from the original boundary.

Comparison Alignment We compared our method with LIVE, DiffVG, O&R, and SGLIVE, under the same number of vectors N (64, 128, 256). For DiffVG, we initialized and optimized N vectors. For O&R, we initialized $4N$ vectors and optimized, then reduced the count to N. For LIVE and SGLIVE, the process involves adding vectors in blocks of increasing sizes, following an order of 8, 8 16, 32, 64, and 128. Vectors are added until the total number of vectors added equals N. Our method prioritized adding structure-wise primitives, up to a maximum of $N/2$, with the remaining count filled by visual-wise vectors to ensure the total reached N.

3. Additional Results

Comparison among Different Image Simplification Methods Gaussian filtering generates four levels of simplified images by varying the kernel size of the Gaussian filter to 2, 6, 10, and 14. Bilateral filtering produces four levels of simplified images by setting the parameters (diameter, σ_{Color}, σ_{Space}) to $(10+5N, 100+50N, 100+50N)$, where $N = 0, 1, 2, 3$. Superpixel algorithm achieves four levels of simplification by reducing the number of superpixels the image is divided into, using values of 400, 200, 100, 50. Figure 3 presents additional results of reconstructed vector layers obtained with these different image simplification methods.

Comparison with Different Vectorization Methods Figure 4 and Figure 5 shows additional results of vector layers constructed by our method and four methods. Figure 6, 7, 8 and 9 compares the difference in visual fidelity and boundaries.

Figure 3. Comparison of vector layers between SDS-based method and three conventional image simplification methods

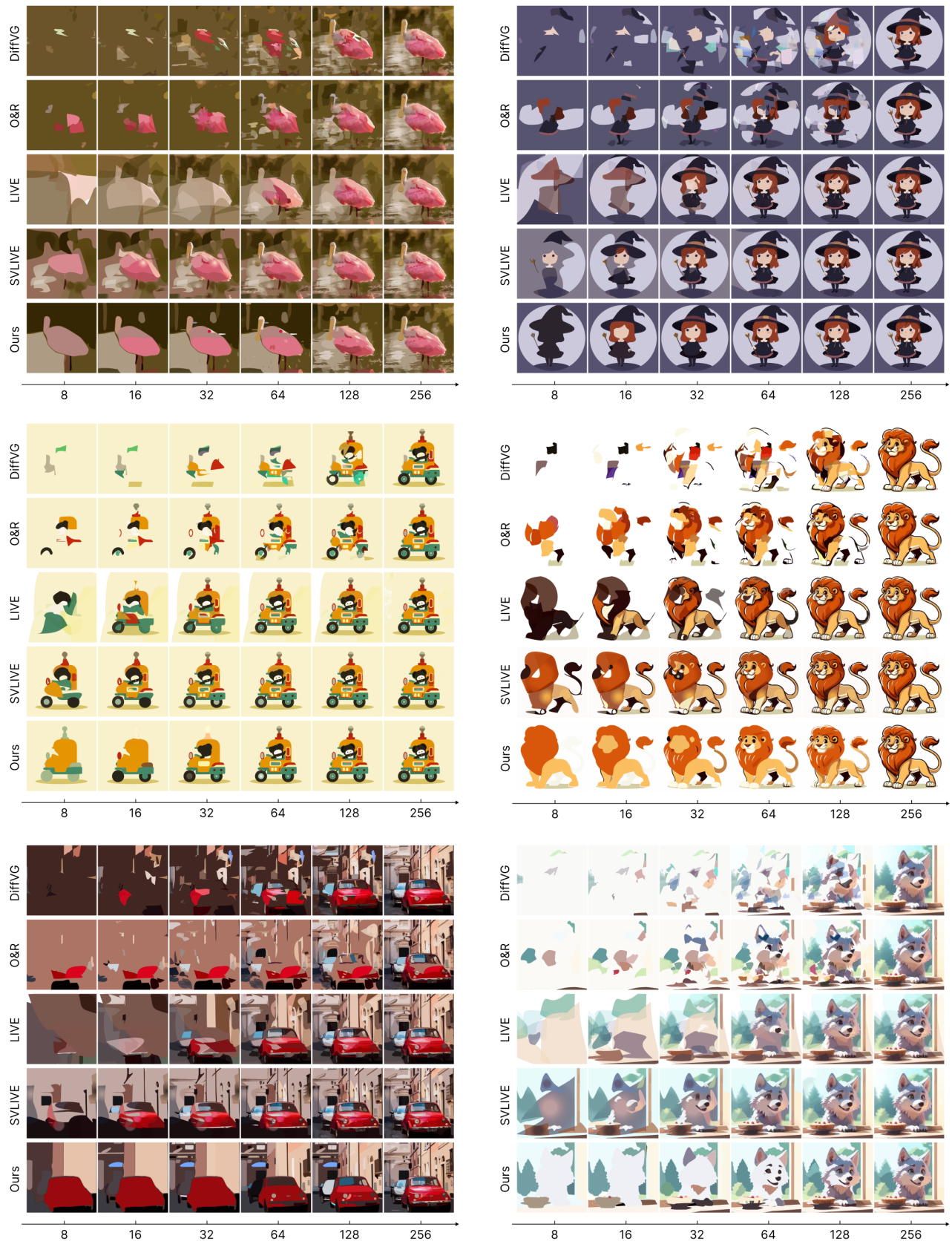

Figure 4. Comparison of vector layers between SDS-based method and three conventional image simplification methods

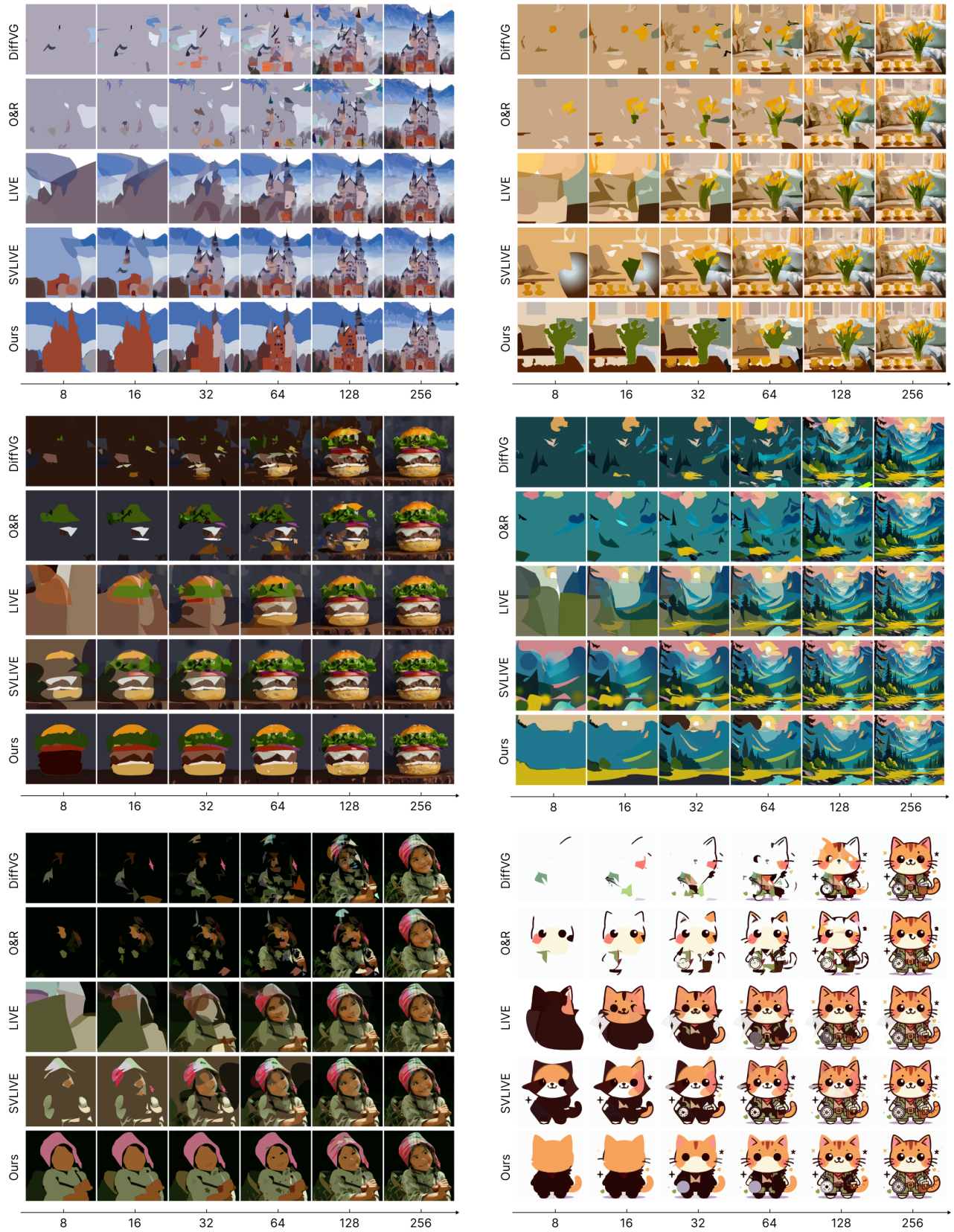

Figure 5. Comparison of vector layers between SDS-based method and three conventional image simplification methods

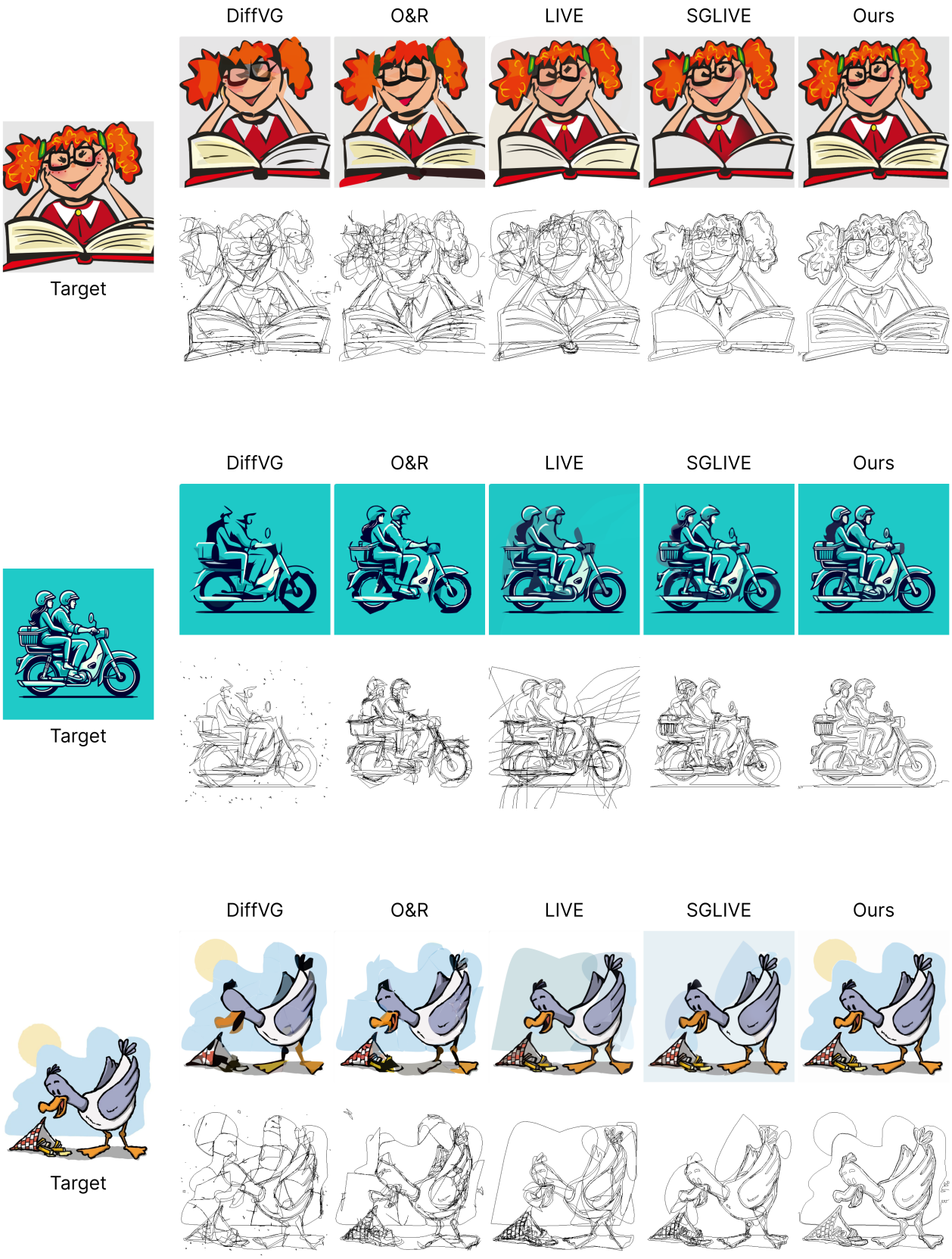

Figure 6. Comparison between our method and four state-of-the-art vectorization methods, vectorized with 128 primitives.

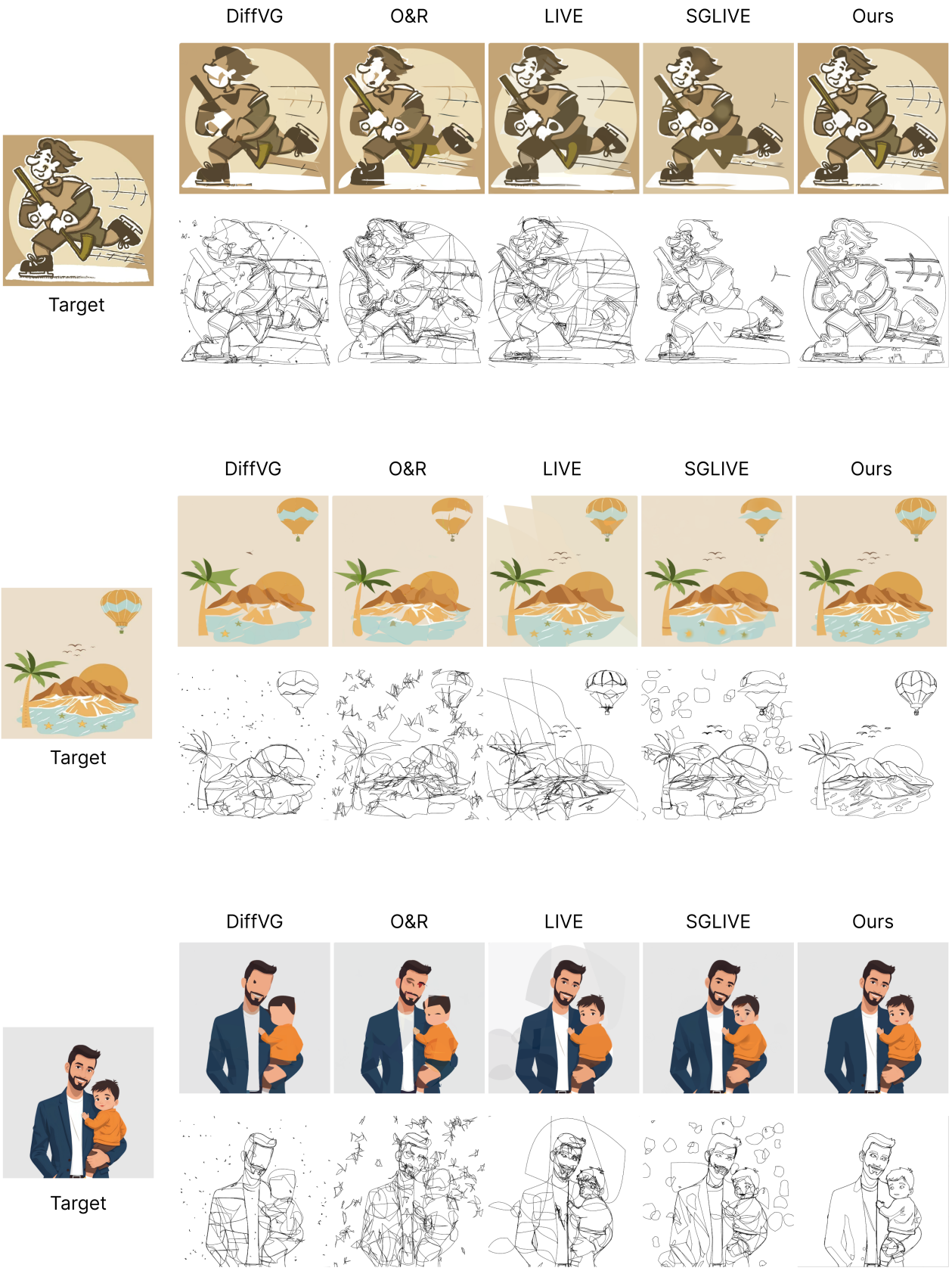

Figure 7. Comparison between our method and four state-of-the-art vectorization methods, vectorized with 128 primitives.

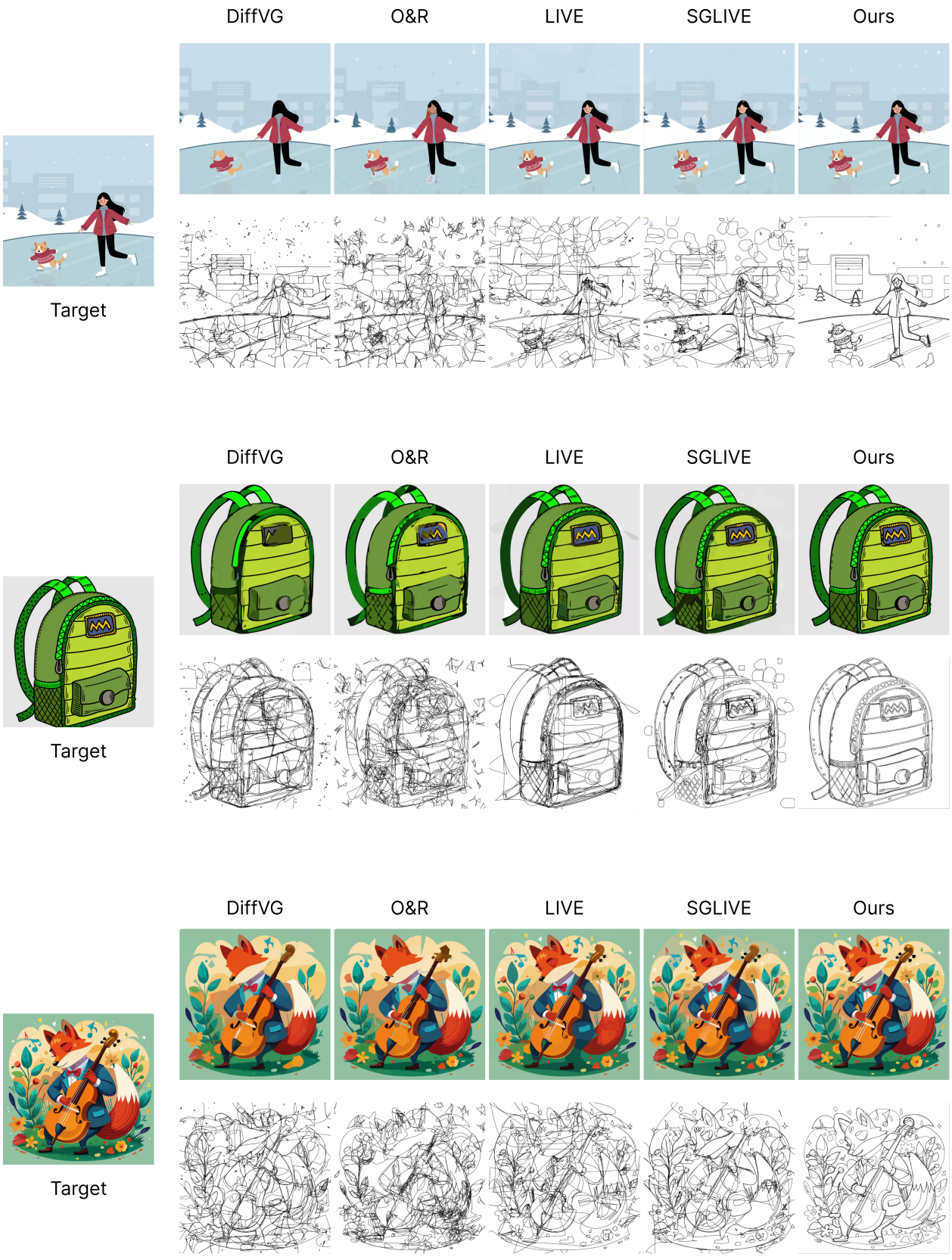

Figure 8. Comparison between our method and four state-of-the-art vectorization methods, vectorized with 256 primitives.

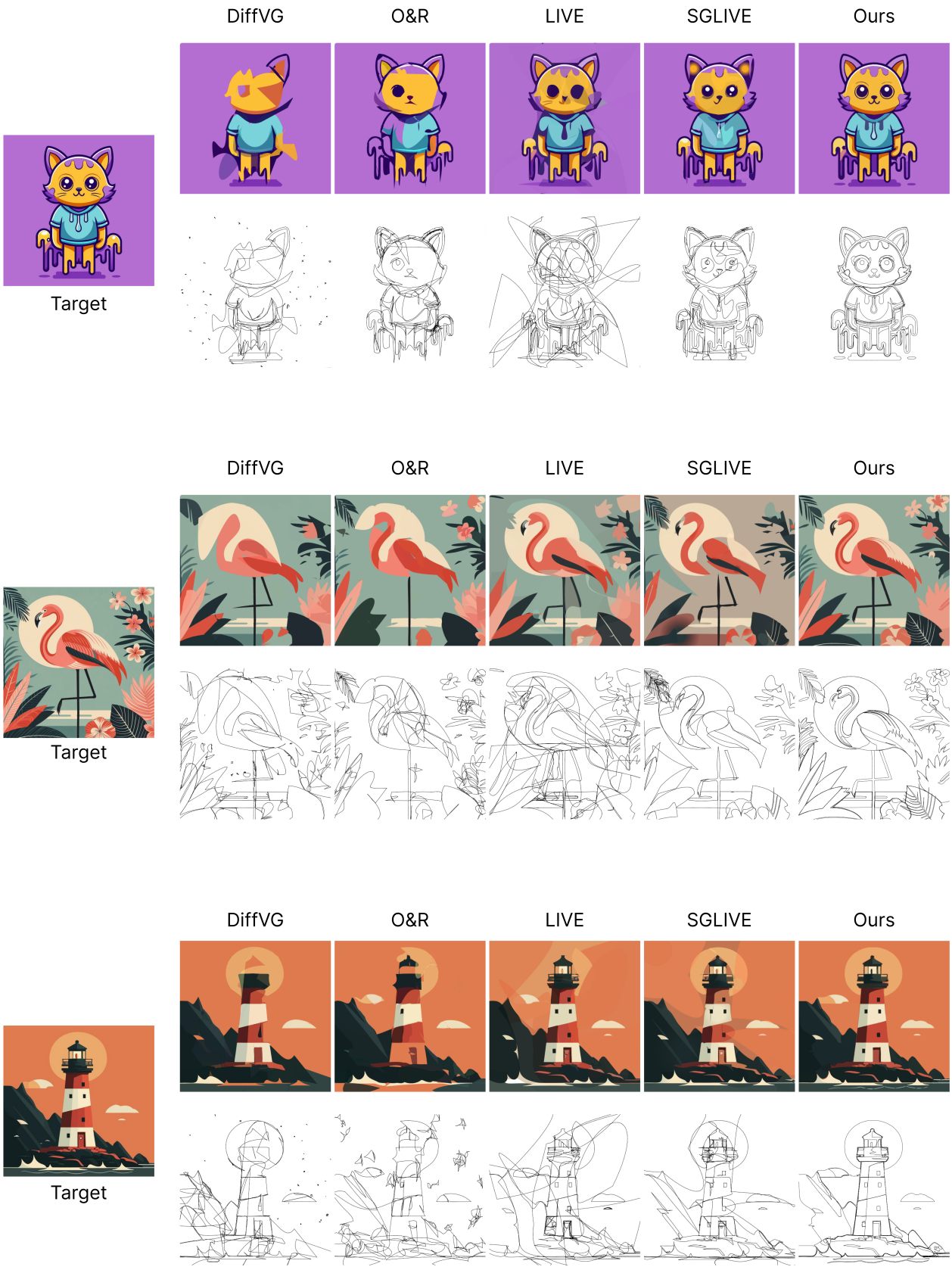

Figure 9. Comparison between our method and four state-of-the-art vectorization methods, vectorized with 64 primitives.