Supplementary Materials for:
DA Wand: Distortion-Aware Selection using Neural Mesh Parameterization

A1. Synthetic Dataset Construction

This section contains more details on how we construct our Near-Developable Dataset. For a given sampled subset of shape primitives (from cube, cone, cylinder tetrahedron, sphere with replacement) normalized to the unit sphere, we loop through each shape primitive and sample a random number of SimpleDeform operations from Blender [8]. The four types of deformations available are [TWIST, BEND, TAPER, STRETCH]. For each deformation operation, we sample a random angle between -0.5 and 0.5 and a random factor quantity from -0.8 to 0.8. Following deformations, we also sample a random axis through the origin and angle and apply a rotation. Finally, we apply a random translation by sampling between -0.3 and 0.3 for each axis. Following the CSG union operation using PyMesh, we apply a 3D augmentation to the resulting shape which involves sampling 1/3 of the vertices of the shape, sliding them by up to 5% in either direction along their normal, and applying a Laplacian Smoothing operation.

We maintain correspondences between the ground truth segmentations of each primitive and the final CSG shape. We use this correspondence to map each ground truth segmentation to a segmentation of the CSG shape. There is no guarantee this mapped segmentation is contiguous due to the CSG, so we identify all connected components of the segmentation. We parameterize each connected component separately using SLIM and compute the resulting isometric and conformal distortion D_I, D_C. If both D_I and D_C are under our threshold of 0.05, then we sample faces from within the segmentation and save them along with the new ground truth segmentation label to generate our labelled data. The full algorithm is shown in Algorithm 1.

A2. Proof of Theorem 1

Theorem 1 *Let $\bar{\mathbf{F}} \subset \mathbf{F}$ be a subset of triangles of the mesh, which comprises one connected component. Let W be non-negative weights assigned to the triangles s.t. the weights of $\bar{\mathbf{F}}$ are non-zero. Let $\mathbf{U}_W$ be the minimizer of Eq. (3) w.r.t W. Then $\mathbf{U}_W$, restricted to $\bar{\mathbf{F}}$, is a well-defined, continuous function of W. Furthermore, if the non-zero weights of W are all equal to 1, then $\mathbf{U}_W$ restricted to $\bar{\mathbf{F}}$ is exactly equal to the (non-weighted) LSCM parameterization of $\bar{\mathbf{F}}$.*

Proof. LSCM's minimizer is uniquely defined up to a global scaling and rotation which can be chosen by fixing two vertices. Thus, WLOG, we pin two vertices of $\bar{\mathbf{F}}$ to two corners of the unit grid. Then, the LSCM minimization problem over $\bar{\mathbf{F}}$ has a unique solution, as proved in [36]. Assume we satisfy the theorem's assumption and all the non-zero weights of W are equal to 1 and are within $\bar{\mathbf{F}}$. Then, for any parameterization $\mathbf{U}$ and its restriction to $\bar{\mathbf{F}}$, denoted $\mathbf{U}|_{\bar{\mathbf{F}}}$, it holds that

$$E_{\text{wLSCM}}(\mathbf{U}) = \sum_{\mathbf{t}\in\mathbf{F}} w_{\mathbf{t}} ||A_{\mathbf{t}} - S(A_{\mathbf{t}})||^2$$
$$= \sum_{\mathbf{t}\in\bar{\mathbf{F}}} ||A_{\mathbf{t}} - S(A_{\mathbf{t}})||^2 = E_{\text{LSCM}}(\mathbf{U}|_{\bar{\mathbf{F}}}),$$

namely any minimizer $\mathbf{U}_W$ of E_{wLSCM} w.r.t the weights W, restricted to $\bar{\mathbf{F}}$, is the unique minimizer of LSCM of $\bar{\mathbf{F}}$.

Furthermore, $\mathbf{U}_W|_{\bar{\mathbf{F}}}$ is a minimizer of a strictly-convex quadratic energy Eq. (3), and hence is the solution of a linear equation, i.e., $\mathbf{U}_W$ is the root of a linear polynomial with coefficients which are linear combinations of W. Since W is non-zero for $\bar{\mathbf{F}}$, $\mathbf{U}_W$ restricted to $\bar{\mathbf{F}}$ is well-defined. Since polynomial roots are continuous in their coefficients, $\mathbf{U}_W$ restricted to $\bar{\mathbf{F}}$ is a well-defined, continuous function of W. □

A3. $\%D_I^\lambda$ with Changing λ and Cost Thresholds

We compare results on the metric % D_I^λ on all 3 datasets for values of λ from 0.01 to 0.1 in Fig. A1. "DCharts v1" refers to DCharts with the default parameters from the main paper."DCharts v2" and "DCharts v3" refer to versions of DCharts with the cost threshold set to 0.1 and 0.3, respectively. For the LogMap baseline, we set the distortion threshold cutoff used for the segmentation heuristic to be equal to λ. We mark $\lambda = 0.05$ with a red line, which is the threshold we report in our main paper.

A4. Postprocessing, Selection Stability, and Timing

DA Wand produces compact segmentation results which are disk topology and near-disk topology. In order to guarantee a disk topology segmentation, we apply a floodfill procedure which takes the largest contiguous subset of the segmented region starting from the selection point. We follow up with a graphcuts procedure to smooth out potential jagged edges along the segmentation boundary, which is a standard procedure in mesh segmentation. We visualize these steps in Fig. A2.

DA Wand also produces highly stable selections on models with sharp feature curves and clear developable regions. Note how in Fig. A3, even when the selection point is on the boundary of the feature curve, our method is able to robustly segment the same developable patch bounded by the feature curve.

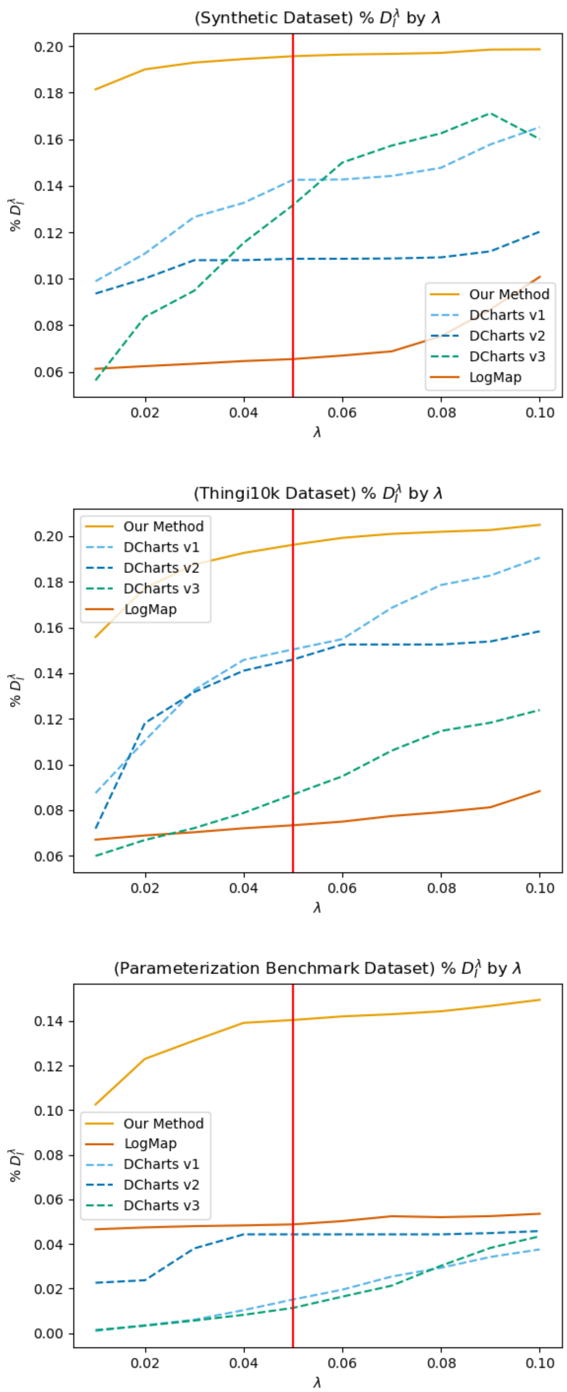

Figure A1. $\%D_I$ metric with sliding λ values. "DCharts v1" refers to DCharts with the default parameters from the main paper."DCharts v2" and "DCharts v3" refer to versions of DCharts with the cost threshold set to 0.1 and 0.3, respectively. The red line marks $\lambda = 0.05$, which is the threshold we report in the main paper.

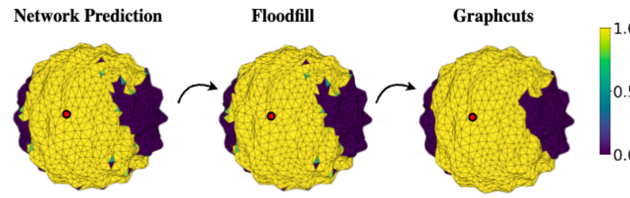

Figure A2. An example of our method's raw prediction and the subsequent effects of the post-processing steps.

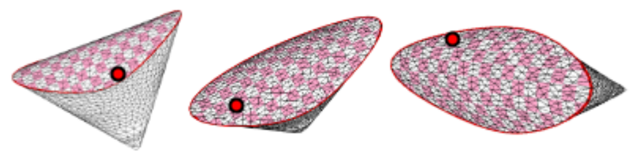

Figure A3. DA Wand is highly stable on selections near sharp feature boundaries which border a developable patch.

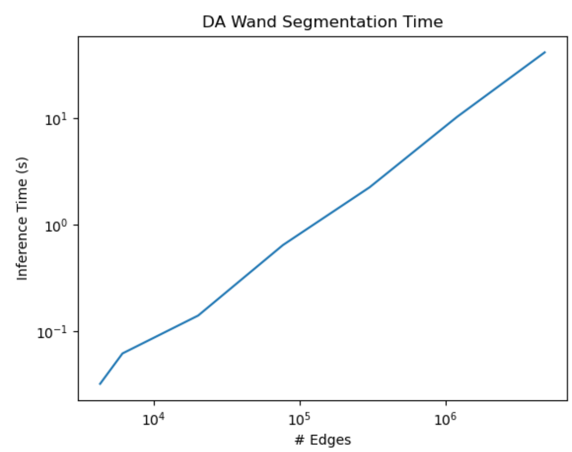

Figure A4. Our method produces segmentations in roughly linear time with respect to the # edges of the model.

Our network architecture builds on MeshCNN, which is a fully convolutional U-Net, implying O(n) time complexity with respect to the input size, in this case the number of edges of the mesh. We show this roughly linear complexity in Fig. A4, where we are able to segment models with 1 million edges in around 10 seconds on a RTX2080 gpu.

A5. Additional Qualitative Comparisons

We show qualitative comparisons between DA Wand and the LogMap and DCharts baseline methods in Fig. A5. Due to the strict distortion threshold and the logarithmic maps' sensitivity to noisy geometry, the LogMap segmentations are low distortion but conservative. On the other hand, DCharts will generally produce large segmentations, but is highly unreliable on natural shapes or shapes with noisy geometry.

We also show qualitative comparisons of the ground truth and DA Wand predictions over the synthetic dataset in Fig. A6. Note that in most cases, the ground truth predictions are constrained to the smallest bounding plane or cylinder of the selection point, whereas our method can segment far beyond nearby feature curves to achieve much larger local parameterizations with little to no cost in distortion.

A6. Parameterization Benchmark: Artist Global Segmentations

We show a few examples of the artist global segmentations which contain our selection points from the Parameterization Benchmark Dataset in Fig. A7. These segmentations were intended for *global UV parameterization*, which involves different priors from *local parameterization*, as explained in Sec. 2. From Fig. A7 it is clear that our segmentations are preferable in the context of local texturing or decaling, as we achieve large segmentations with little to no tradeoff in terms of parameterization distortion.

Algorithm 1 Near-Developable Shape Generation

procedure DEFORM(M, n)
  for i in range(n) **do**
    $\theta \leftarrow$ Unif$(-0.5, 0.5)$
    $\alpha \leftarrow$ Unif$(-0.8, 0.8)$
    method = randomChoice(['TWIST', 'BEND', 'TAPER', 'STRETCH'])
    $M \leftarrow$ blenderSimpleDeform(method, θ, α)
  end for
end procedure
procedure RANDOMROTATION(M)
  $v \leftarrow$ Unif$(0, 1,$ size $= 3)$
  $\theta \leftarrow$ Unif$(0, 2\pi)$
  rotate(M, v, θ) ▷ rotate m about axis v by θ
end procedure
procedure RANDOMTRANSLATION(M)
  M.vertices $\leftarrow$ Unif$(-0.3, 0.3,$ size $= 3)$
end procedure
procedure AUGMENT(M)
  $vlen \leftarrow$ len(M.vertices)
  $vi \leftarrow$ randomChoice(range(vlen), size=vlen/3)
  M.vertices$[vi] \leftarrow M$.vertices$[vi]$ + M.vertexnormals$[vi]$ $*$ Unif$(0, 0.05,$ size=length(vi))
  LaplacianSmooth(M)
end procedure
procedure GENERATESHAPE(primitives, segs)
  for M in primitives **do**
    deform(M, randomChoice(range(3,10)))
    randomRotation(M)
    randomTranslation(M)
  end for
  M', correspondences $\leftarrow$ csgUnion(primitives)
  $M' \leftarrow$ augment(M')
  for M, mmap, seg in zip(primitives, correspondences, segs) **do**
    mappedseg $\leftarrow$ mmap(seg)
    mappedseg $\leftarrow$ collect all connected mapped segmentation components
    for mseg in mappedseg **do**
      $uv \leftarrow$ SLIM(mseg)
      $ss \leftarrow$ getSingularValues(uv)
      $d_I \leftarrow$ mean$((\max(ss[:,0], 1/ss[:,1]) - 1)^2)$
      $d_C \leftarrow$ mean$((ss[:,0] - ss[:,1]))^2)$
      if $d_I \leq 0.05$ and $d_C \leq 0.05$ **then**
        Sample initial selection faces f in mseg, and save inputs (M', f) and corresponding ground truth label mseg
      end if
    end for
  end for
end procedure

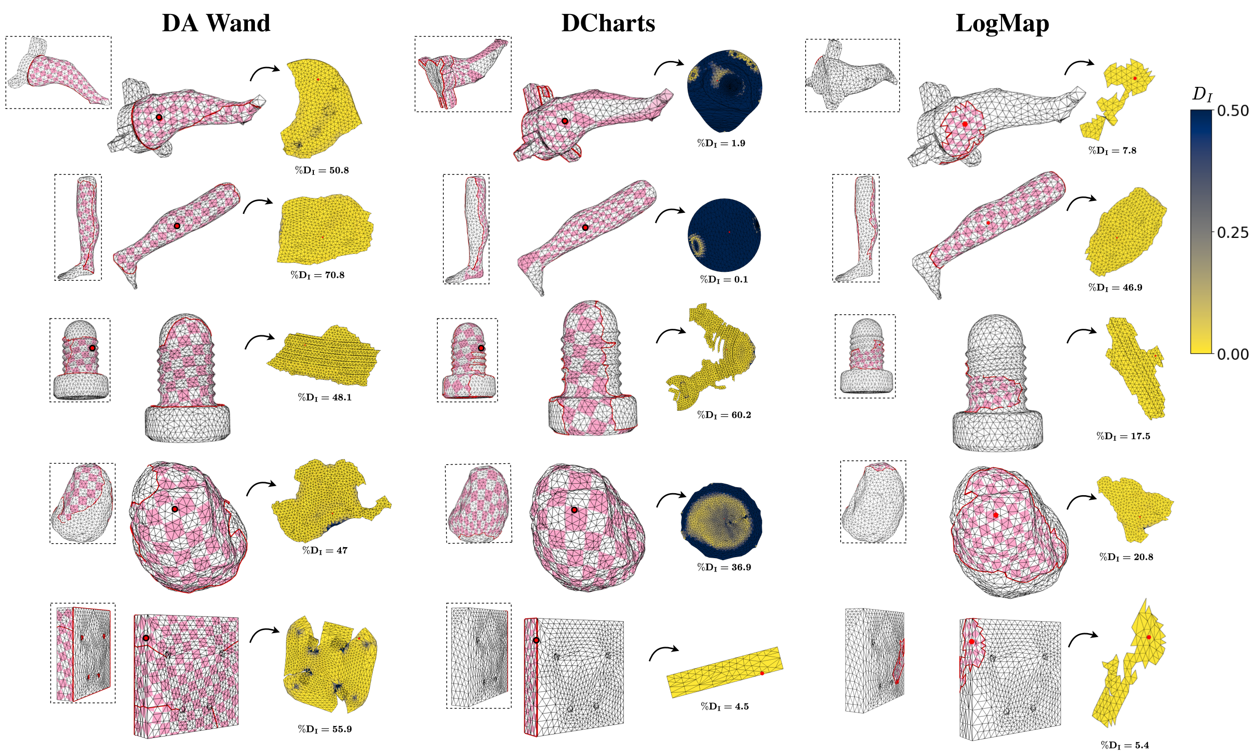

Figure A5. Sample segmentations between DA Wand, DCharts, and the LogMap baselines.

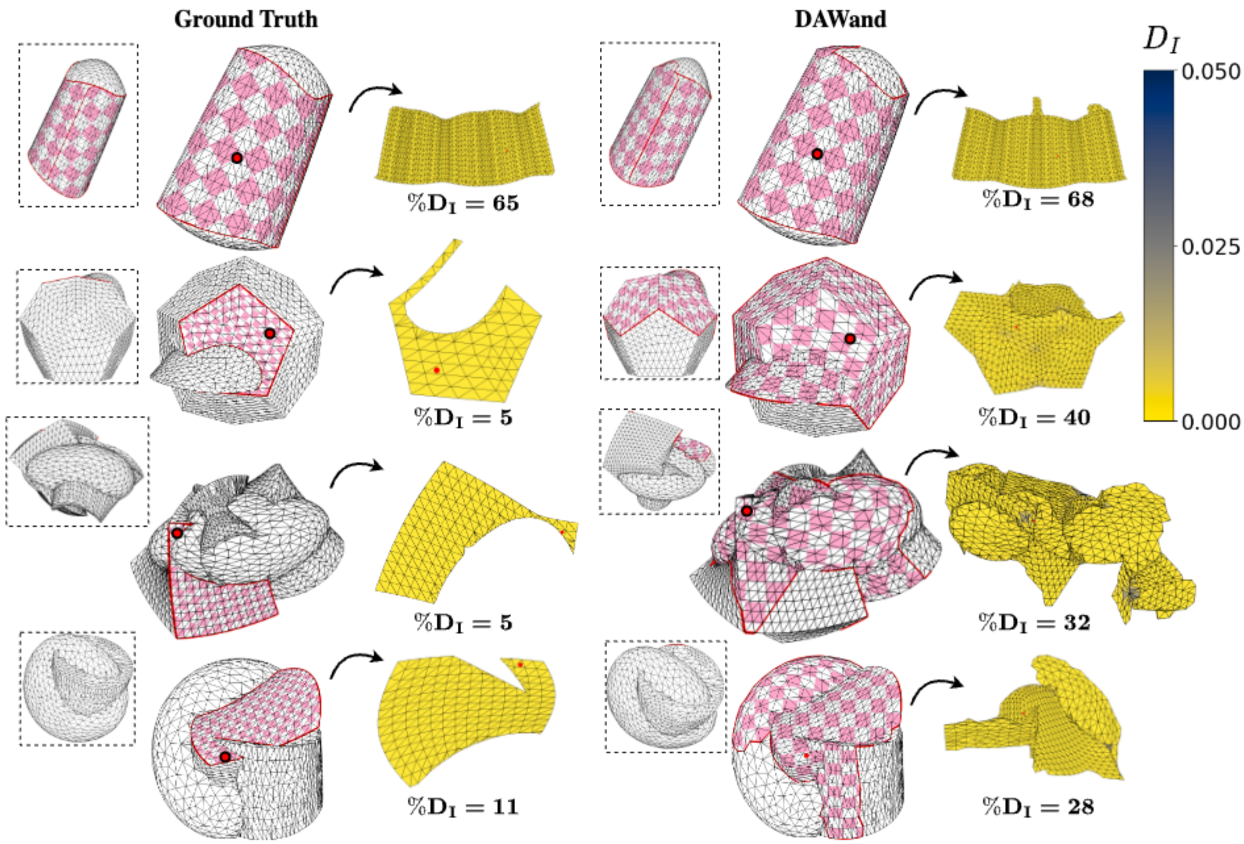

Figure A6. Sample segmentations between DA Wand and the ground truth over the synthetic test set. We report the percentage of triangles under isometric distortion 0.05 $\%D_I$ in bold.

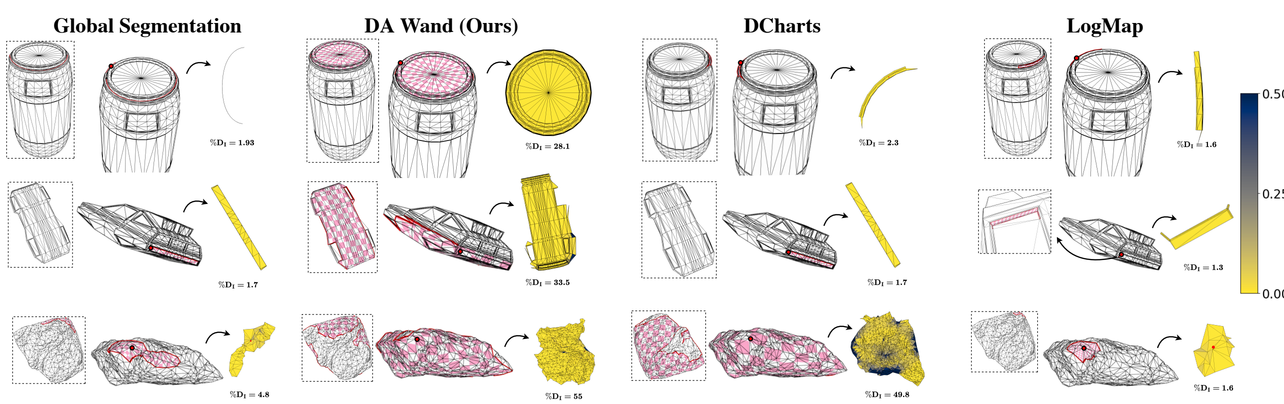

Figure A7. Sample segmentations between the Parameterization Benchmark labels (Global Segmentation), DA Wand, DCharts, and the LogMap baselines.