GFPACK++: Attention-Driven Gradient Fields for Optimizing 2D Irregular Packing - Supplementary Material

Tianyang Xue[1] Lin Lu[1*] Yang Liu[2] Mingdong Wu[3] Hao Dong[3]
Yanbin Zhang[1] Renmin Han[1] Baoquan Chen[3]
[1]Shandong University [2]Microsoft Research Asia [3]Peking University

The supplementary materials include visualizations and detailed information, such as the complexity of polygons in each dataset and statistics from both training and testing.

1. Dataset Complexity

The distribution of our training data is described in Table 1. The distribution of vertex numbers in the polygons of each dataset is presented in Table 2 and Fig. 1. Every polygon in the Puzzle dataset is unique.

Table 1. Teacher dataset utilization distributions.

Teacher dataset	Min	Avg	Max
Garment	63.02%	72.39%	78.90%
Dental	67.33%	74.16%	77.36%
Atlas (building)	39.92%	74.55%	95.74%
Atlas (object)	24.69%	63.25%	85.18%
Atlas (general)	40.76%	67.54%	92.41%

Therefore, we randomly sampled 1000 polygons from the Puzzle dataset to calculate the vertex counts. The number of polygons in each dataset is displayed in Table 3. We also provide visualizations of the polygons in the 'Datasets' folder.

Table 2. Polygon Vertex counts.

Dataset	Min	Avg	Max	Std
Garment	5	81.57	284	76.83
Dental	17	34.47	123	14.50
Puzzle	4	8.25	29	2.86
Atlas (building)	4	6.52	70	3.87
Atlas (object)	4	67.41	1120	102.38

*Corresponding author: llu@sdu.edu.cn

Table 3. Polygon counts in each dataset.

Dataset	Polygon Count
Garment	313
Dental	440
Atlas (building)	3262
Atlas (object)	6764

2. Impact of Enhancement

The enhancement process, which integrates the constrained gradient optimization (as described in Sec. 4.4 of the main text) and gap elimination (detailed in Sec. 5.2 of the main text), is employed to resolve minor overlaps and eliminate gaps. The impact of enhancement on other algorithms is not significant (about 1%, see Table 4), as they do not contain overlaps, have relatively small gaps, and offer little room for local optimization.

Table 4. Average improvements achieved through enhancement.

Dataset	GFPACK++		XAtlas		SVGnest	
	Bef.	Aft.	Bef.	Aft.	Bef.	Aft.
Garment	74.25%	77.04%	68.90%	69.53%	70.98%	72.75%
Dental	75.11%	77.53%	69.88%	70.86%	72.55%	73.64%

3. Training Settings

We conducted our training and testing on a Linux server equipped with 4 NVIDIA RTX 4090 GPUs and an Intel(R) Xeon(R) Platinum 8163 CPU. GFPACK++ converged in 50 hours. GFPACK converged after 168 hours of training. We continued training for an additional 48 hours after convergence for both methods.

The factor 10 in Eq. 11 of main text amplifies the sigmoid input, effectively increasing the weight contrast between high- and low-utilization samples. This helps the network better focus on high-quality data. Empirically, we found that larger factors tend to suppress low-quality sam-

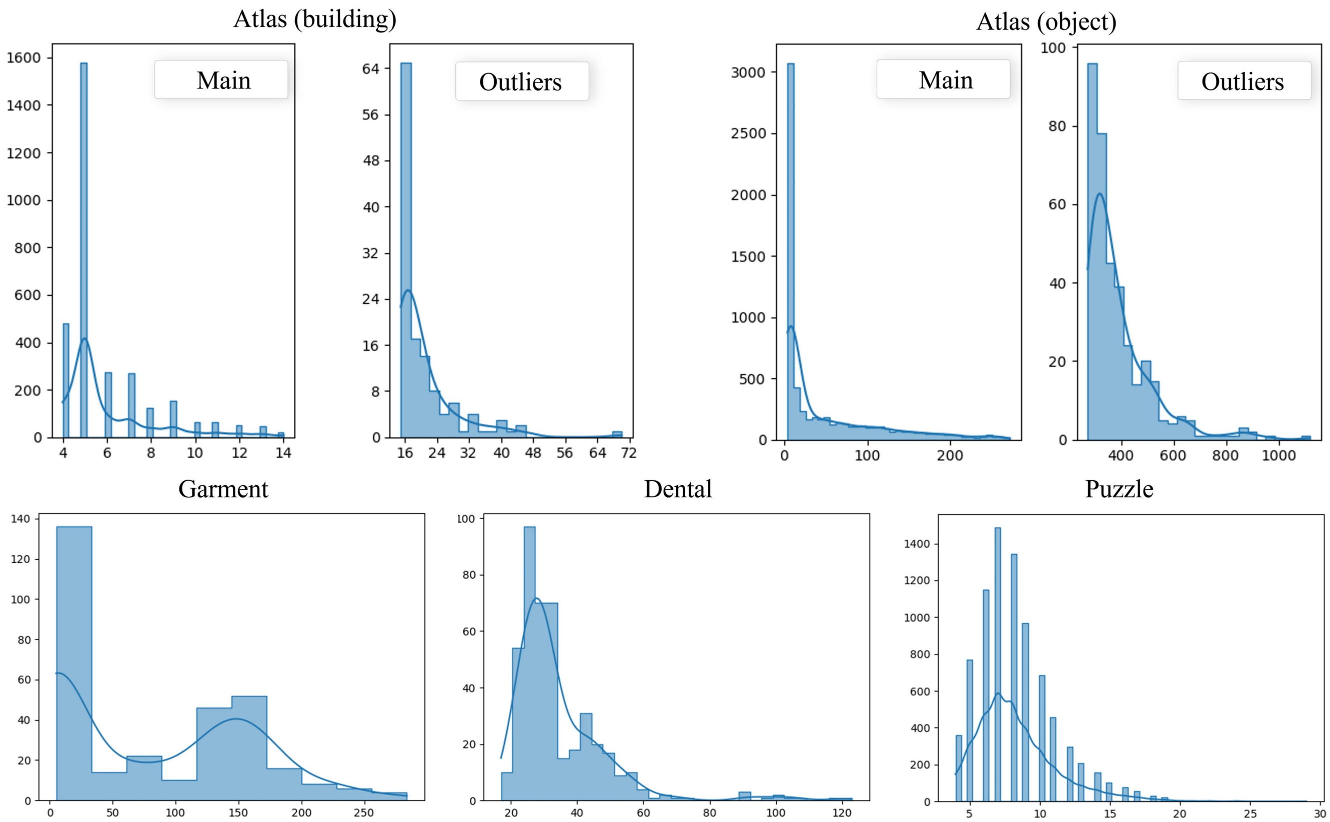


Figure 1. The statistics of polygon vertex counts. The horizontal axis represents the number of vertices in a polygon, while the vertical axis indicates the frequency of these vertex counts. Due to the wide distribution of vertex counts in the Atlas dataset, we have divided it into two parts: the main distribution and the outliers.

ples too aggressively, while smaller ones fail to sufficiently distinguish useful data. A value of 10 achieves a good balance and yields the best overall performance.

We quadrupled the number of parameters in GFPACK from 10 M to match GFPACK++ at 40 M. This increase resulted in excessive VRAM usage and extended convergence times, with the model not converging within 240 hours. Despite this, GFPACK still failed to produce collision-free results on datasets with rotations. We did not further increase the number of training parameters since the training time was already too long.

4. Baseline Algorithms Configuration

We modified the code of XAtlas to enable its application to non-atlas packing problems. Key modifications include: (a) the introduction of a packing height constraint to facilitate strip packing operations; (b) adjustments to padding, alignment, and scaling methods to prevent distortion. Detailed XAtlas settings are provided here:

```
options.pack.height = height;
options.pack.rotation = true;
options.pack.rotateChartsToAxis = align;
options.pack.bruteForce = true;
options.pack.padding = 0.5f;
options.pack.bilinear = true;
options.pack.blockAlign = false;
options.scale = 1.0f;
```

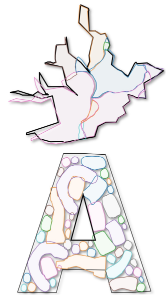
In this context, ‘height‘ specifies the stripe height, with a setting of -1 indicating unfixed boundaries. XAtlas supports rotations of 0 and 90 degrees when both brute force and rotation are enabled. The align parameter determines whether polygons are aligned to their bounding boxes. We have disabled this setting for the dental dataset, while it is enabled for other datasets.

Additionally, for SVGNest, we have enhanced its parallel efficiency to improve its performance within the same time.

5. Validation Visualizations

We present visualizations of validation results (Sec. 6.1 of the main text) generated by GFPACK++ across various datasets (Fig. 2). Additional visualizations are available under the 'Validations' directory.

We also observed failure cases in two typical scenarios: (1) **highly complex boundaries or polygon shapes**, which may lead to conflicting gradient fields and hinder convergence; and (2) **extreme overcrowding**, where the number of polygons far exceeds the layout capacity, leading to inevitable overlaps. Two such examples are shown in the right inset.



6. Comparative Visualizations

We visualized comparative results from GFPACK++, SVGnest, and XAtlas applied to Garment, Dental, Puzzle, and Atlas datasets (Fig. 3). We also visualized comparative results from GFPACK++, [1], SVGnest, and XAtlas applied to Atlas datasets (Fig. 4). The results of [1] are taken directly from the paper. Additional visualizations are available under the 'Validations' directory.

References

[1] Zeshi Yang, Zherong Pan, Manyi Li, Kui Wu, and Xifeng Gao. Learning based 2d irregular shape packing. *ACM Trans. Graph.*, 42(6), 2023. 3, 5

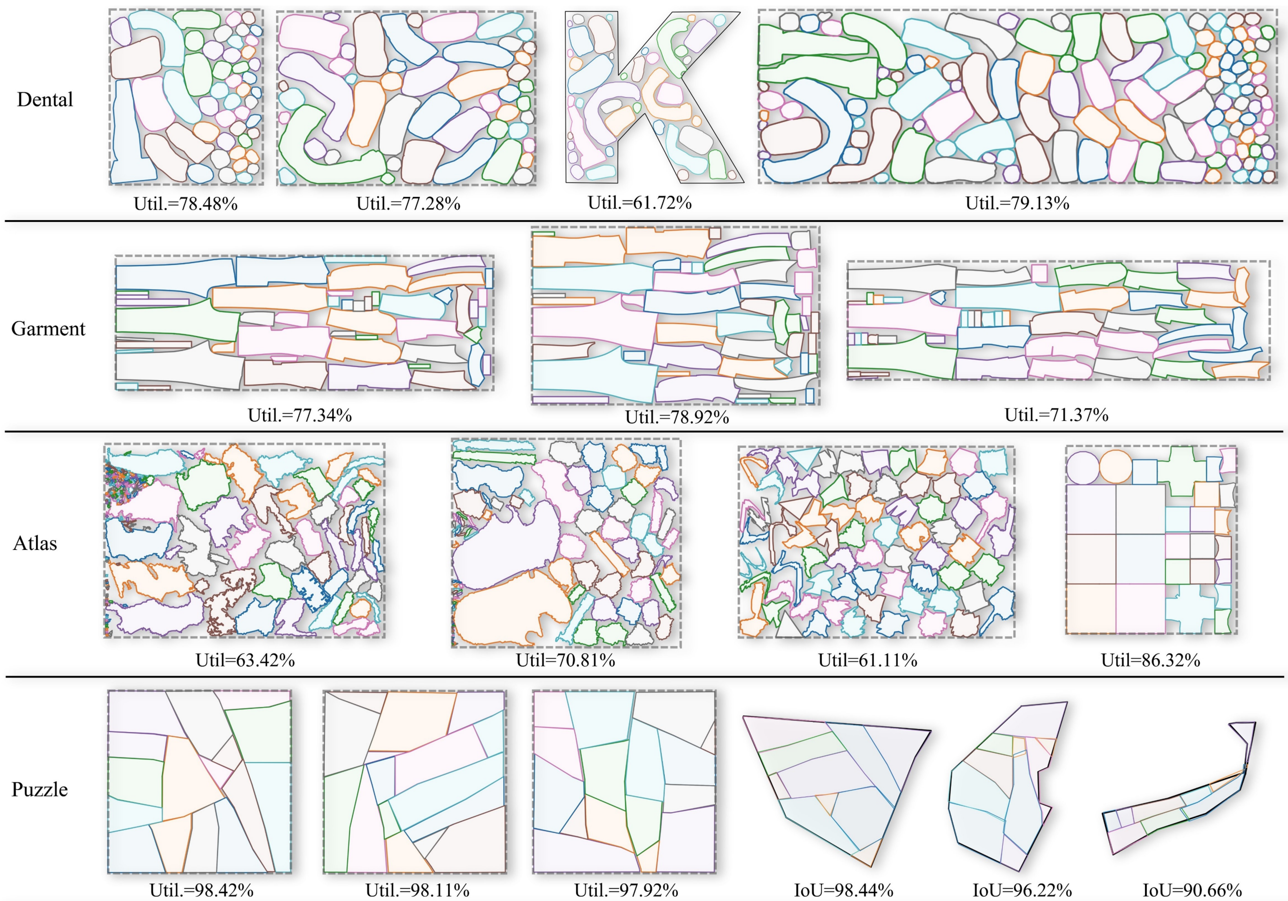


Figure 2. Some results from GFPACK++ across different datasets. Note that the utilization enhancement only applies to regular boundaries, so small overlaps may be observed in the puzzle (arbitrary) results.

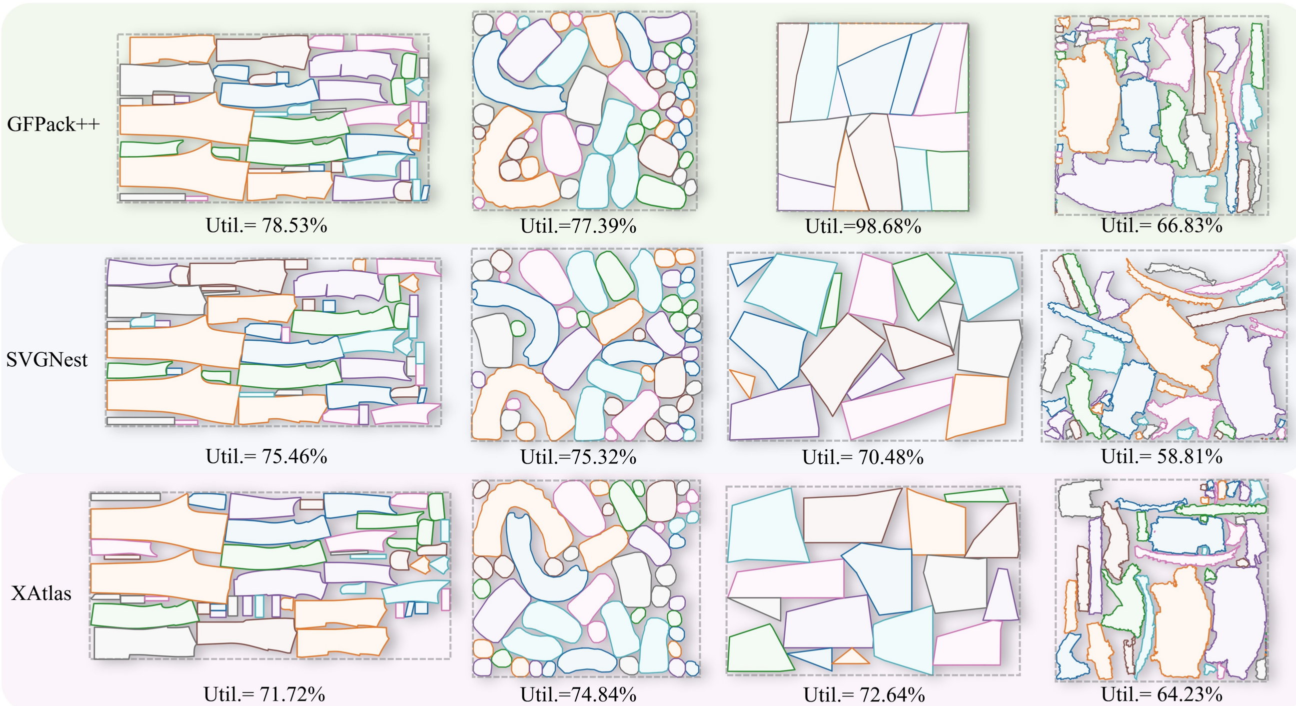


Figure 3. Comparative results from GFPACK++, SVGnest, and XAtlas applied to Garment, Dental, Puzzle, and Atlas datasets. In the lower left corner of the Garment dataset, a shared similar local layout pattern between GFPACK++ and SVGnest can be observed. For the Puzzle dataset, GFPACK++ outperforms the others, nearly achieving a global optimum due to its continuous rotation solving.

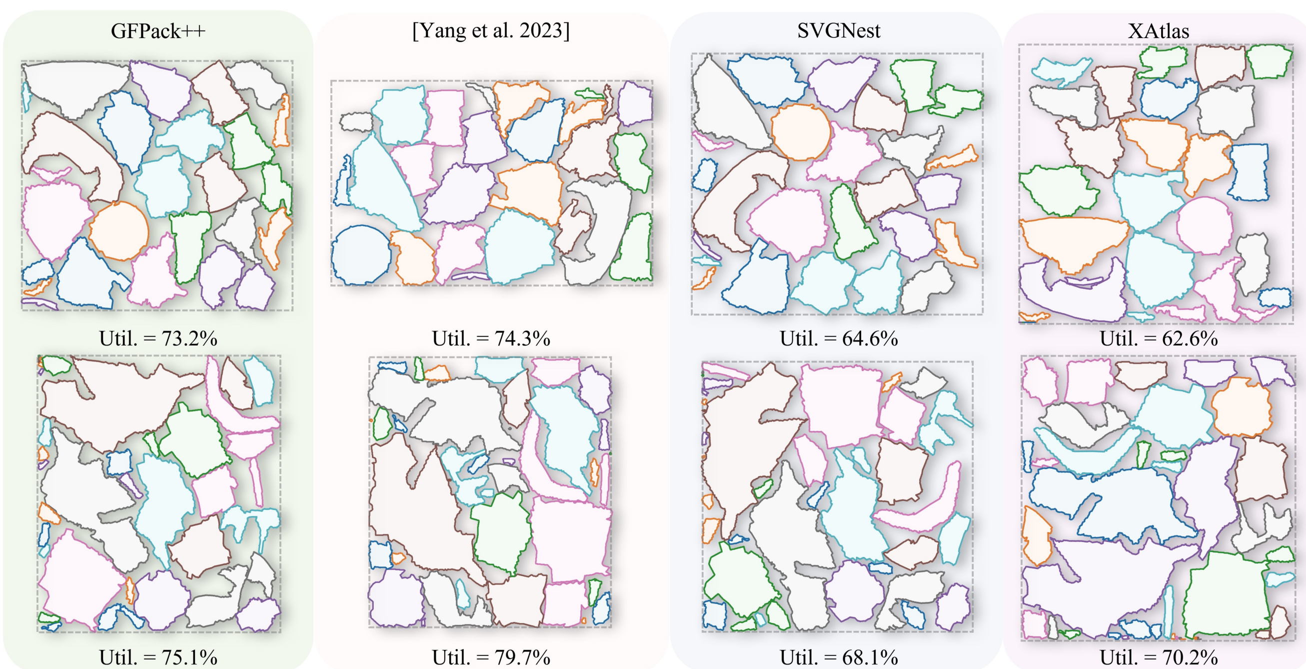


Figure 4. Comparative results from GFPACK++, [1], SVGnest, and XAtlas applied to Atlas datasets.