Towards Layer-wise Image Vectorization
Supplementary Materials

Xu Ma[1], Yuqian Zhou[2,3*], Xingqian Xu[2*], Bin Sun[1],
Valerii Filev[4], Nikita Orlov[4], Yun Fu[1], Humphrey Shi[2,4]
[1]Northeastern University, [2]UIUC, [3]Adobe Research, [4]Picsart AI Research (PAIR)

1. User Study

we conducted an user study* to quantitatively compare our LIVE with DiffVG and Neural Painting. We randomly selected 21 images from emoji and pics datasets, and invited 20 users to select the method that has the best learning processing. The average scores of DiffVG, Neural Painting, and our LIVE are 14.3%, 11.9%, and 73.8%, respectively. Results indicate that most people believe LIVE can achieve the best layer-wise representation.

2. More Qualitative Comparisons

We also tested on more emoji images, as shown in Figure 1. Without bells and whistles, LIVE explicitly disentangles the visual concepts, where each new path can fit a particular component in the input image. Adding more paths would not decrease the performance.

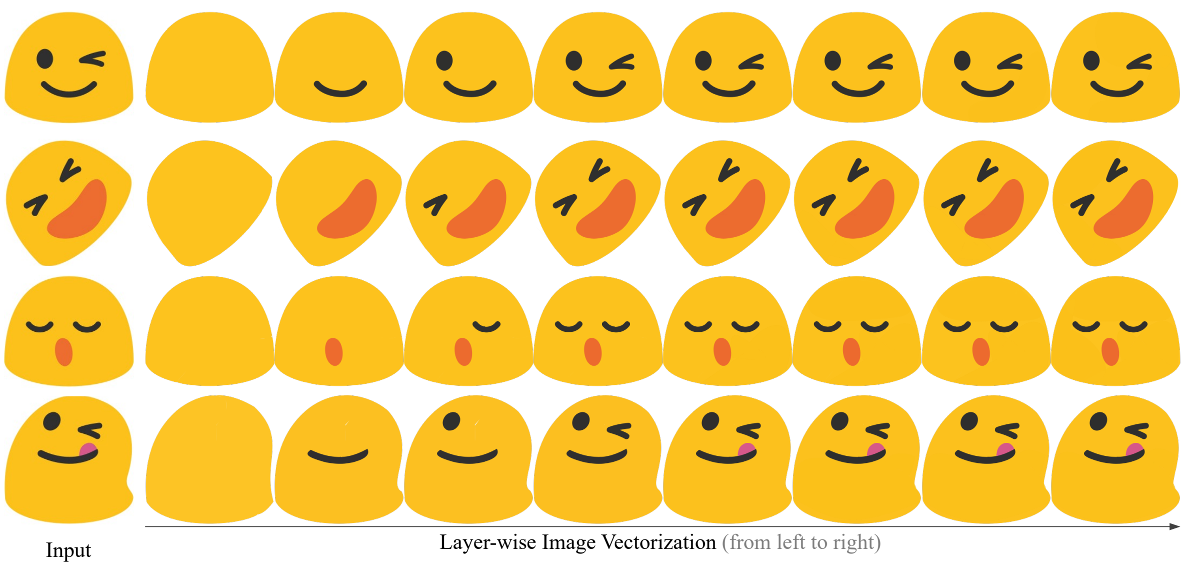


Figure 1. More examples of layer-wise representation. Given a simple image, our LIVE is able to learn each component in the image in a layer-wise fashion. Here we show the learning progress using 8 paths, where each output appends a new path to the previous result.

3. Xing loss weights

We evaluate the impact of Xing loss in Figure 2. Generally, adding the Xing loss would greatly reduce the risk of self-interaction problems. We notice that a small weight of Xing loss can achieve the best result, while a larger weight (*i.e.*, 1.0)

*The details of the user study can be found at: https://wj.qq.com/s2/9665341/19ed.

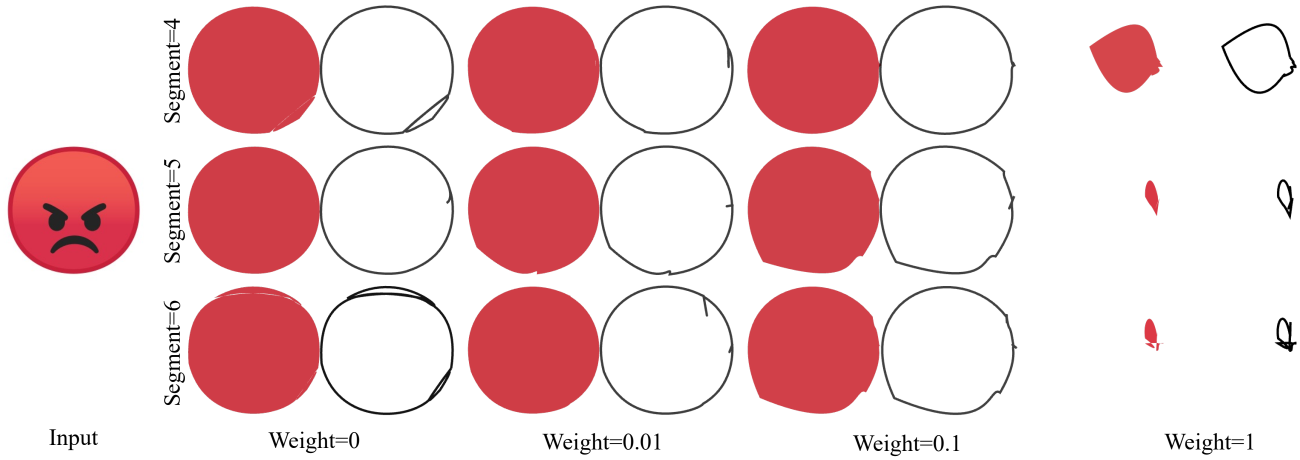


Figure 2. Impact of Xing loss weight. We showcase the results of first generated path with different segments number for illustration. In each pair, left is the generated SVG output and right is the corresponding bézier curve.

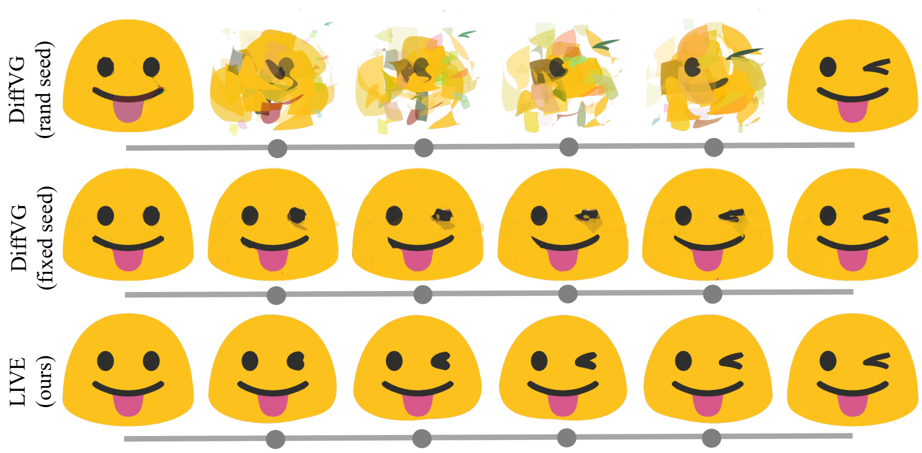


Figure 3. Linearly interpolating two SVGs generated by DiffVG (random seed), DiffVG (fixed seed), and our LIVE.

always leads to the failure of optimization. Empirically, we set the Xing loss weight to 0.01 by default.

4. Orderness of LIVE-generated SVGs

An interesting property of LIVE-generated SVGs is the deterministic order of the optimized bézier paths, due to the progressively learning pipeline and our component-wise initialization method. We demonstrate this property by linearly Interpolating two generated SVGs. We compare the results of DiffVG with rand seed, DiffVG with fixed seed, and our LIVE. Clearly, the interpolation results of DiffVG (with rand seed) are messed up because the path order is not deterministic. Even we fixed all randomness seeds, DiffVG still performs worse than our LIVE.

5. More Interpolation Results

We next present more interpolation results in Figure 4. Rather than interpolating between two images, we further interpolate new images among four randomly selected images. Holistically, the results shown in Figure 4 indicate that combining a VAE model with our LIVE method can achieve similar results as other VAE-based vectorization methods.

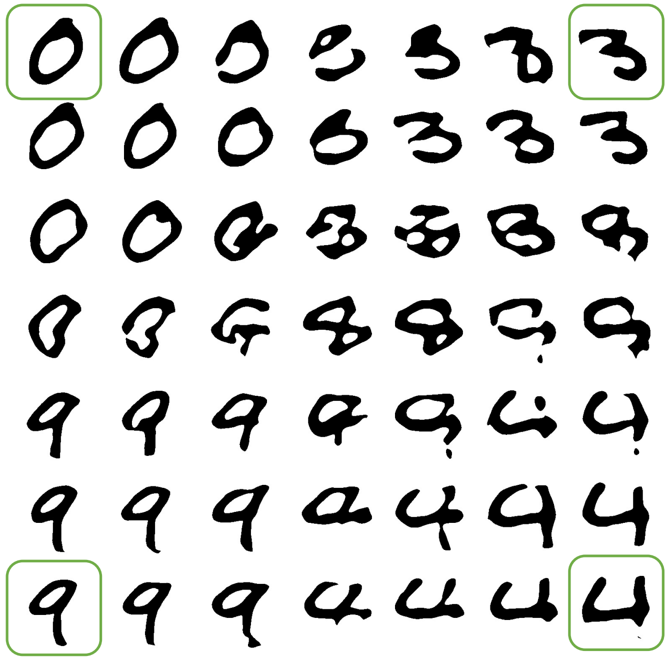
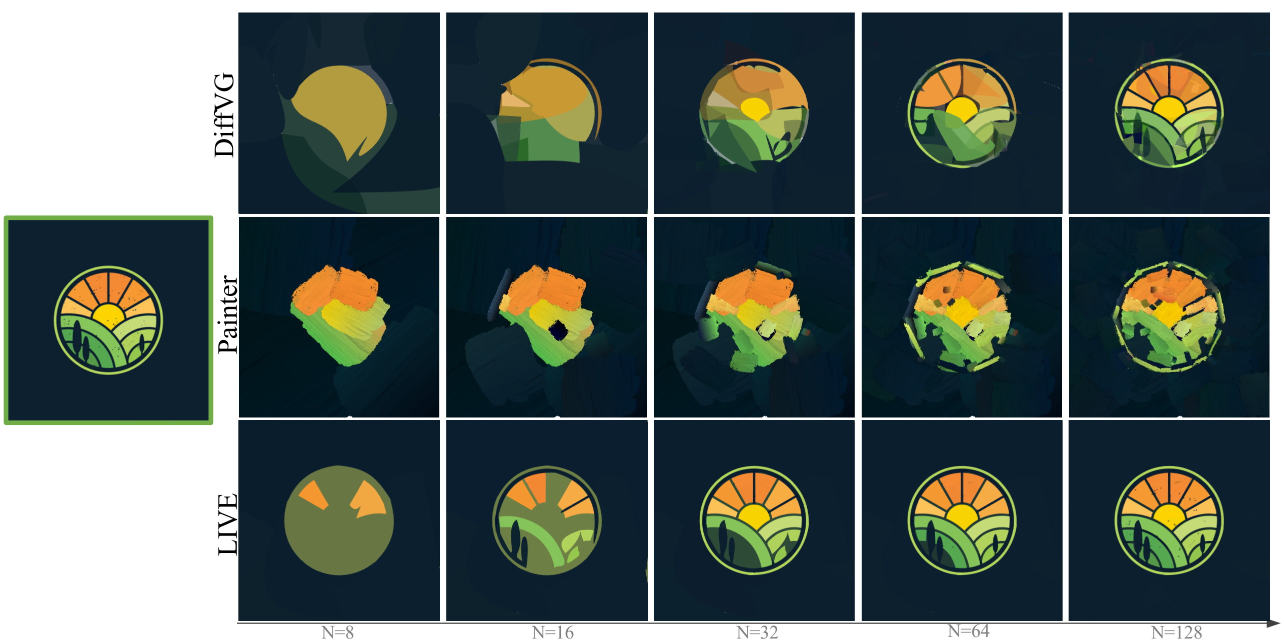


Figure 4. We plot the linear Interpolation results among four digits from MNIST dataset. We use the green boxes to emphasize the raster images. All the rest images are interpolated SVGs generated by the combination of VAE model and our LIVE method.

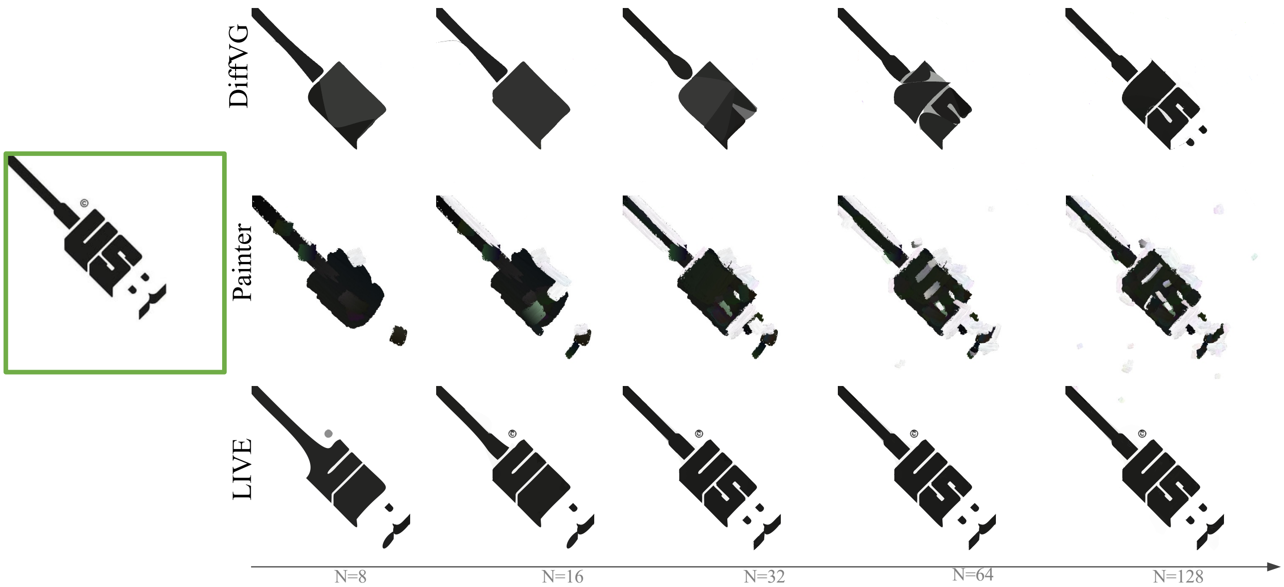
6. More Vectorization Quality

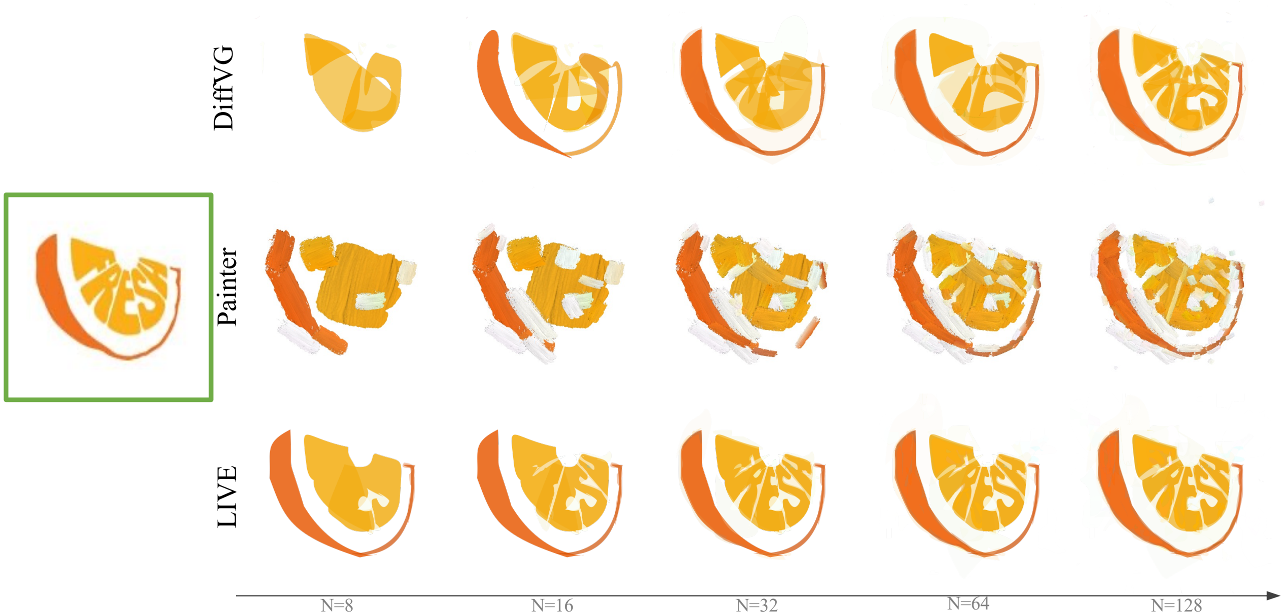
In this section, we present more examples that compare our LIVE with DiffVG and Neural Painting. The results are presented in the following figures, left (in the green box) is the input images, right is the output of different methods. Empirically, under the same conditions (*i.e.*, the number of paths/strokes), our LIVE can exhibit much better representation results, especially when the path number is small. Please zoom in to see the details.

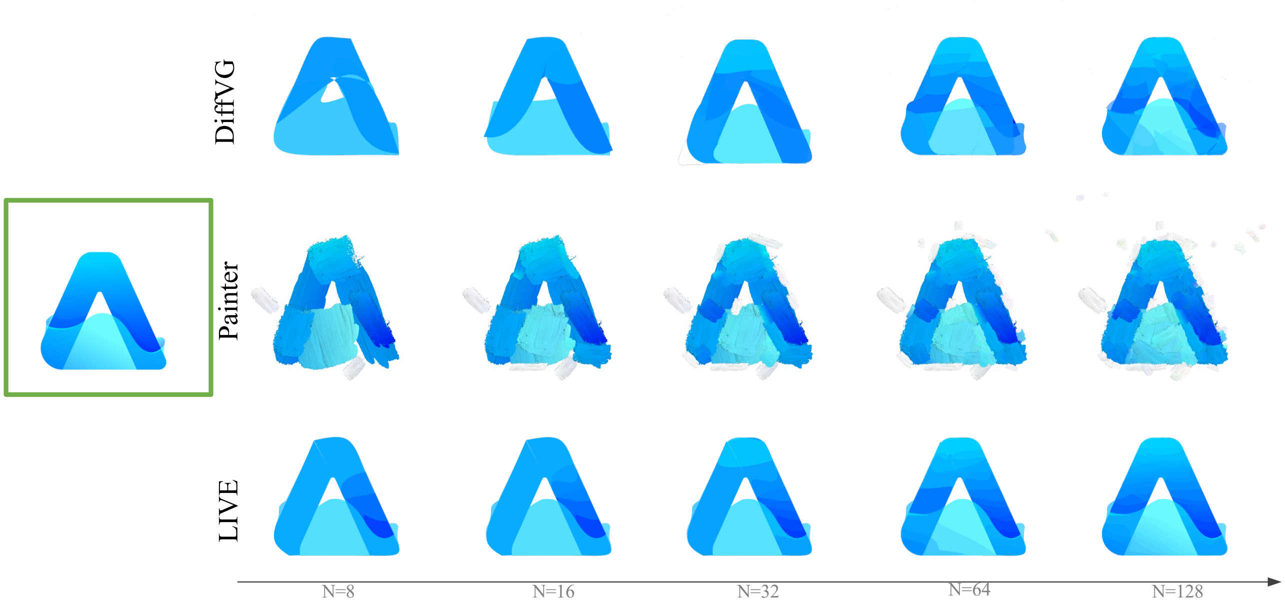



DiffVG
Painter
LIVE
N=10
N=20
N=30
N=40
N=50

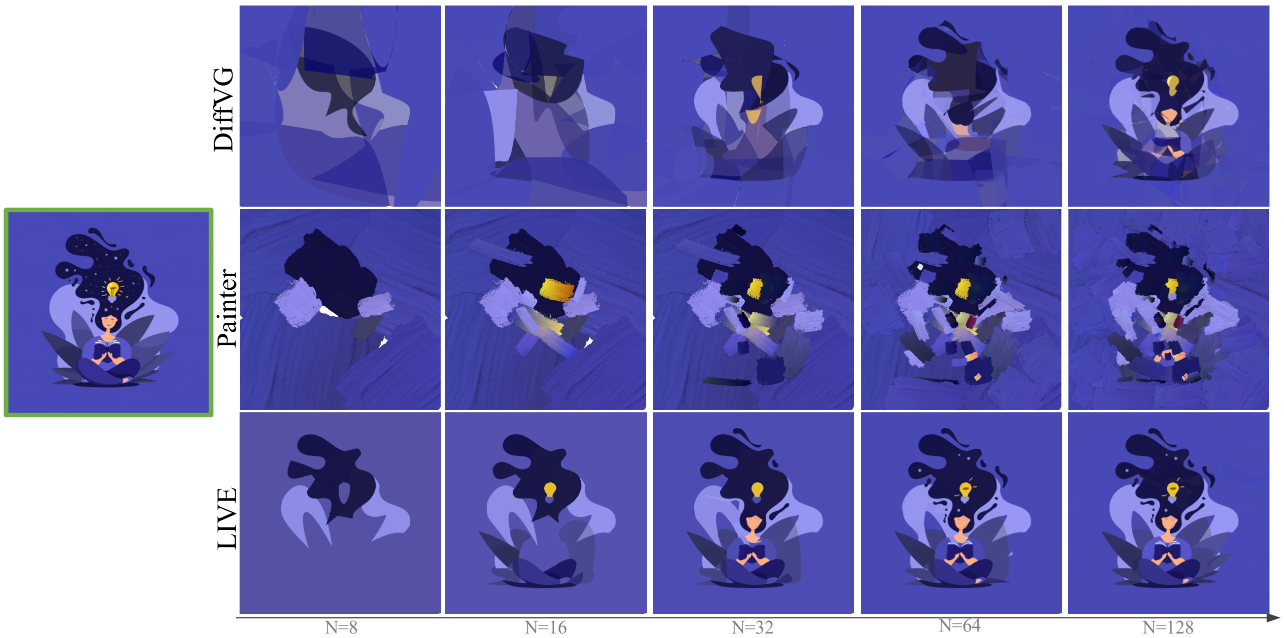

DiffVG
Painter
LIVE
N=16
N=32
N=64
N=128
N=256


DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128

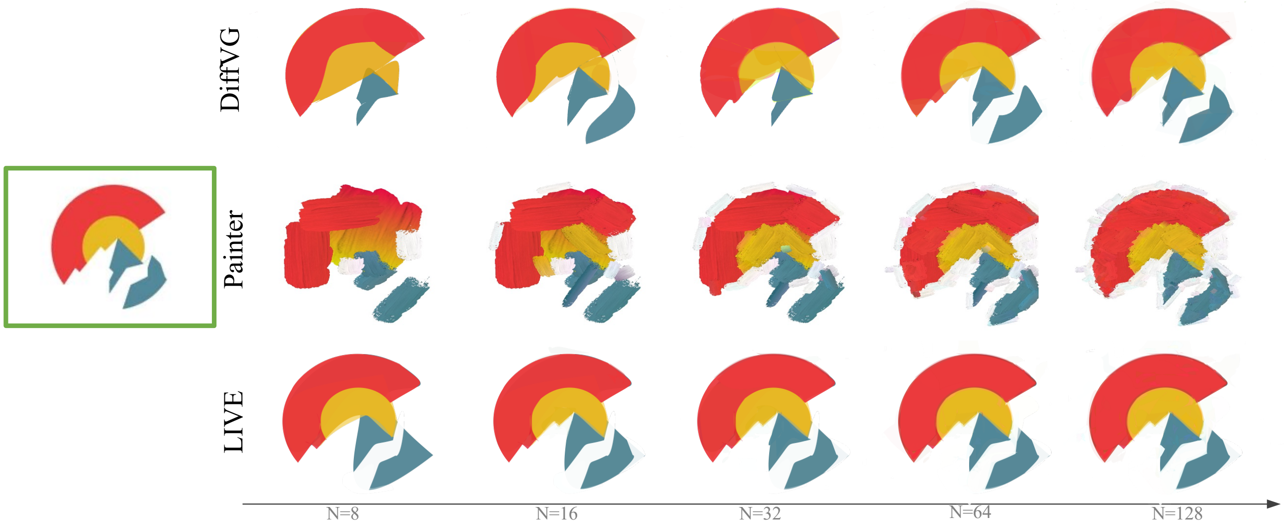

DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128

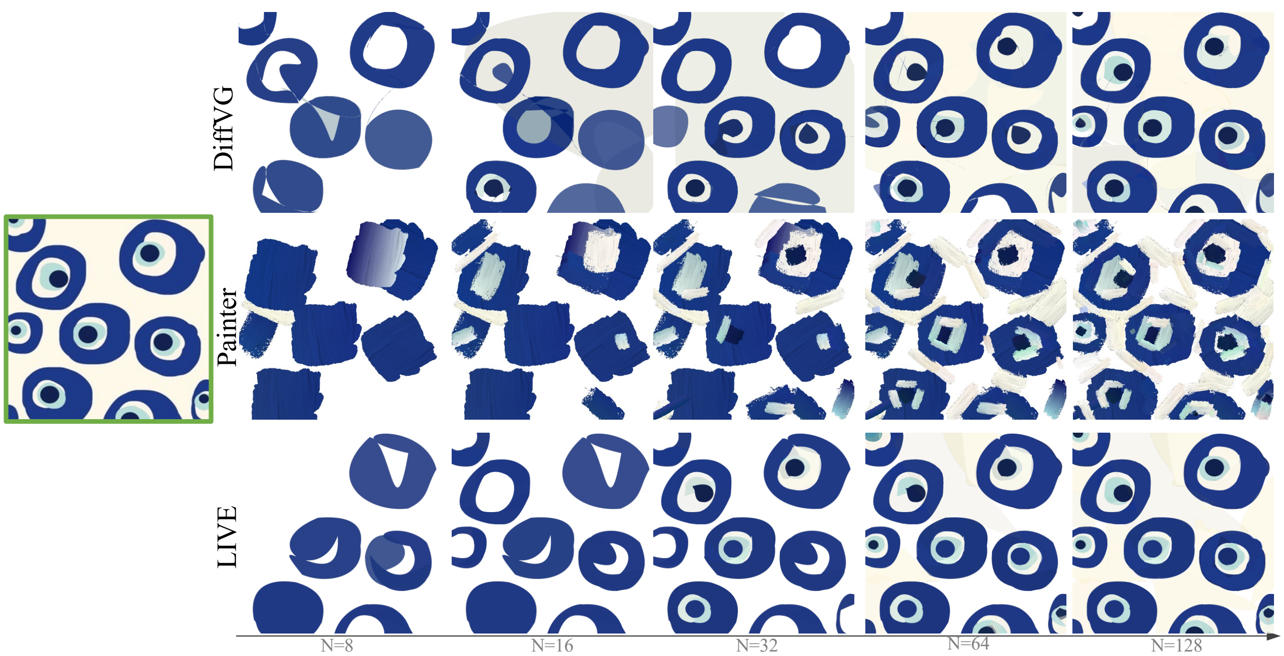

DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128

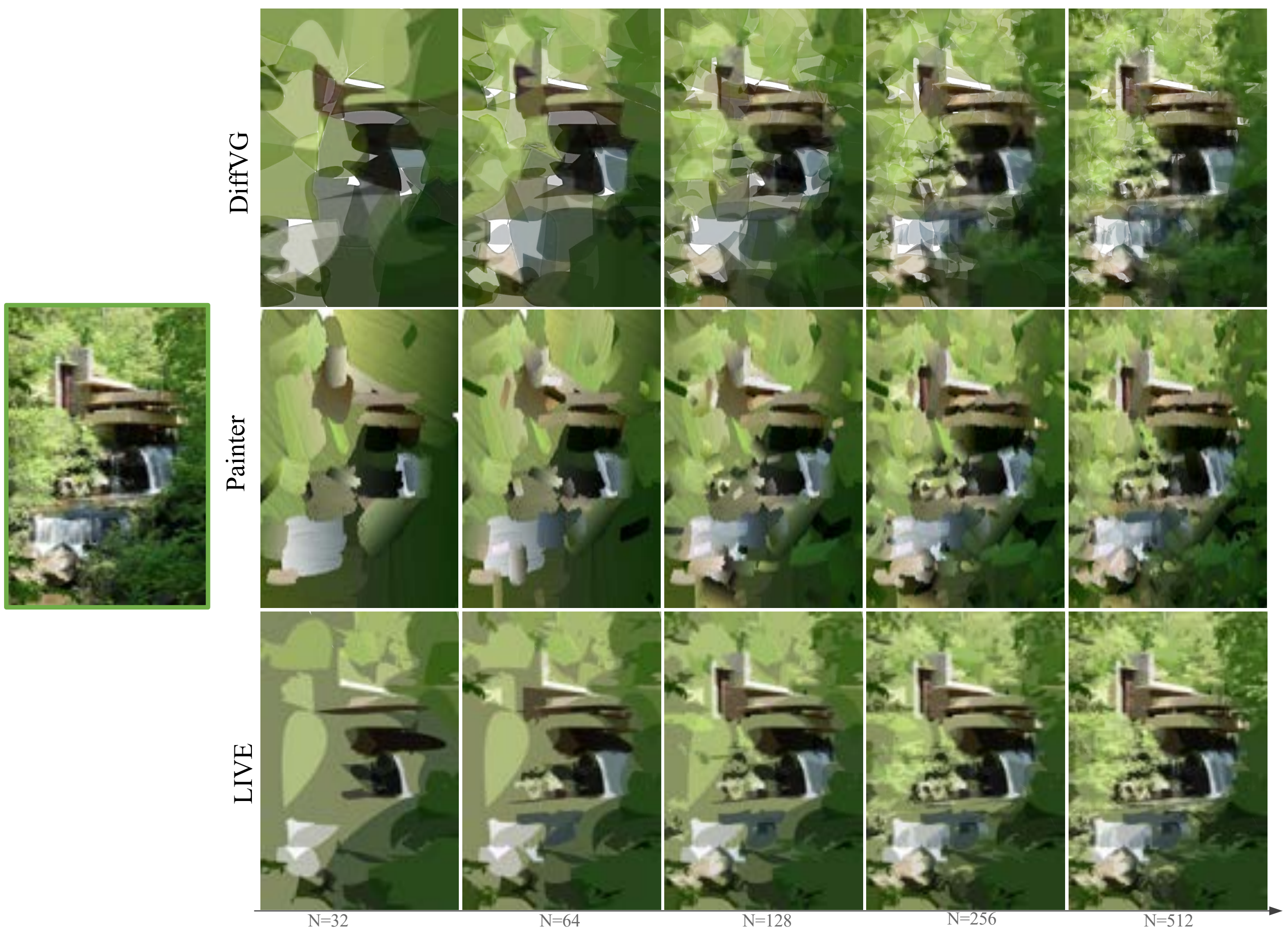

DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128


DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128


DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128


DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128


DiffVG
Painter
LIVE
N=8
N=16
N=32
N=64
N=128


DiffVG
Painter
LIVE
N=32
N=64
N=128
N=256
N=512