

# Im2Vec: Synthesizing Vector Graphics without Vector Supervision

Anonymous CVPR submission

Paper ID 16

## S.1. Network Architecture

Our Encoder network contains 5 2D convolution residual blocks, with [32, 64, 128, 256, 512] filters respectively. All the convolution layers have kernel size 3, stride 2 and zero pad the input by 1 pixel in both spatial dimensions. The convolutional layers are followed by two parallel fully-connecter layers that each output a vector of size 128; they represent the mean and variance for the latent embedding. Our Path decoder has 6 1D convolution layers with [170, 340, 340, 340, 340, 2] channels. All the 1D convolutions have kernel size 3, stride 1 and circular padding of the input by 1 tap. Our auxiliary network contains 4 fully-connected layers with [256, 256, 256, 3] channels, respectively. The sample deformation network contains 3 1D convolution layers with [340, 340, 1] channels, all the convolution layers have the same kernel size, stride and padding as the 1D convolution layers in path decoder. All layers are followed by ReLU activations, except the last layer of the path decoder which is followed by a sigmoid activation.

## S.2. Chamfer Distance

An alternative, commonly used metric to measure the reconstruction accuracy for geometric objects is the Chamfer distance. For two point sets  $X, Y$ , the Chamfer distance is defined as:

$$\sum_{x \in X} \min_{y \in Y} \|x - y\|^2 + \sum_{y \in Y} \min_{x \in X} \|x - y\|^2. \quad (1)$$

In Table S1, we show the bidirectional Chamfer distance computed between the synthesized and ground truth geometries, with points sampled uniformly along the shape boundaries (according to the path parameterization). Unlike the pixel-space metrics we report in the main paper, this evaluation suggests our model underperforms the baselines. This is misleading. As shown by Smirnov *et al.* [1], the Chamfer distance varies wildly, depending on the sampling pattern (and the parameterization, by extension). This adversely impacts our method. The baselines (DeepSVG and SVG-VAE) are optimized to regress the ground truth vector parameterization, which leads to a lower Chamfer loss, despite

Table S1: **Reconstruction quality.** Comparison of Bidirectional Chamfer Distance reconstruction losses for various methods and datasets.

	FONTS
ImageVAE	×
SVG-VAE	0.168
DeepSVG	0.136
Im2Vec (Ours)	0.279

worse perceptual fidelity. Conversely, our method is trained on raster data only. It does not seek to retrieve the ground truth parameterization of the curve, but rather to faithfully capture the (rasterized) appearance. Therefore, our model achieves higher fidelity, despite a higher Chamfer distance.

## S.3. Auxiliary Network

In Figure S1b, we plot the reconstruction error vs. number of path segments for 5 randomly sampled instances from the FONTs dataset, as estimated by our trained Im2Vec model. In Figure S1c, we show a plot of the reconstruction error vs. the number of segments modelled by our auxiliary network. We use Equation (2) to select the appropriate sampling rate for new shapes generated using our method. This enables us represent each of the generated shape in the most compact way. We mark ‘x’ in the Figure S1c for  $k = 0.005$ .

**N = number of curves and k = Rate of change of loss.**

$$\text{loss} = c + \exp(b - a * N) \quad (2)$$

$$d\text{loss}/dN = -a \exp(b - a * N) \quad (3)$$

$$\Rightarrow k > -a \exp(b - a * N) \quad (4)$$

$$\Rightarrow k/a < \exp(b - a * N) \quad (5)$$

$$\Rightarrow \log(k/a) < b - a * N \quad (6)$$

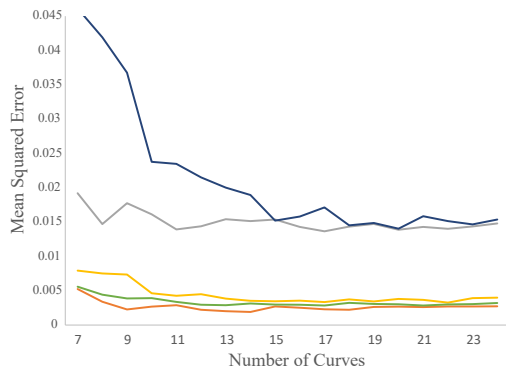
$$\Rightarrow -b + \log(k/a) < -a * N \quad (7)$$

$$\Rightarrow -\log(k/a) + b > a * N \quad (8)$$

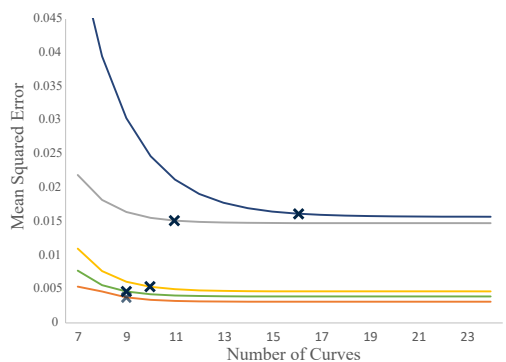
$$\Rightarrow \frac{\log(a/k) + b}{a} > N \quad (9)$$



(a) visual fidelity vs. number of segments



(b) error vs. number of segments; ground truth



(c) error vs. number of segments; modelled

Figure S1: **Auxiliary network output.** Our auxiliary network helps us choose the best sampling density on the unit circle, such that we express the generated shape with the fewest number of Bézier curves based on the user defined reconstruction error threshold.

#### S.4. Robustness to Inconsistent Correspondence

In Section 3.4 of the main paper, we describe a pipeline that segments the sub-components of a design or font, using off-the-shelf tools, which improves the interpretability and consistency of latent space interpolations. In Figure S2, we show that differential compositing makes our method robust to potential inconsistencies in this automatic labelling step. For this experiment, we specifically created training and test datasets of font character ‘8’ where the openings are colored inconsistently. Note that our model still manages to capture meaningful interpolations for instances that have consistent labels.

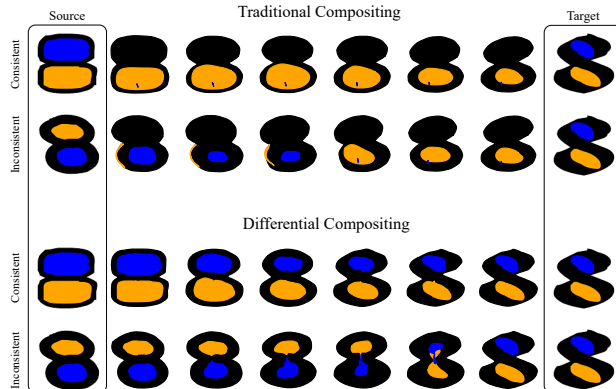


Figure S2: **Training on inconsistent dataset.** We show latent space interpolations using models trained with traditional compositing versus differential compositing. In both the scenarios, we show examples of interpolations between consistently labelled instances and inconsistently labelled instances. When trained with differential compositing, in both the scenarios our model is robust to pre-processing inconsistencies.

#### References

- [1] D. Smirnov, M. Fisher, V. G. Kim, R. Zhang, and J. Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 561–570, 2020. 1